IBM VisualAge für Java Version 4.0
Professional - Release-Hinweise

Professional - Release-Hinweise

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter 'Bemerkungen' gelesen werden.

- Die IBM Homepage finden Sie im Internet unter: ibm.com
- IBM und das IBM Logo sind eingetragene Marken der International Business Machines Corporation.
- Das e-business Symbol ist eine Marke der International Business Machines Corporation
- Infoprint ist eine eingetragene Marke der IBM.
- ActionMedia, LANDesk, MMX, Pentium und ProShare sind Marken der Intel Corporation in den USA und/oder anderen Ländern.
- C-bus ist eine Marke der Corollary, Inc. in den USA und/oder anderen Ländern.
- Java und alle Java-basierenden Marken und Logos sind Marken der Sun Microsystems, Inc. in den USA und/oder anderen Ländern.
- Microsoft Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.
- PC Direct ist eine Marke der Ziff Communications Company in den USA und/oder anderen Ländern.
- SET und das SET-Logo sind Marken der SET Secure Electronic Transaction LLC.
- UNIX ist eine eingetragene Marke der Open Group in den USA und/oder anderen Ländern.
- Marken anderer Unternehmen/Hersteller werden anerkannt.

Zweite Ausgabe (Juni 2001)

Diese Ausgabe bezieht sich auf Version 3.5.3 of IBM(R) VisualAge for Java.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM VisualAge® for Java™, Version 4.0 Professional Release Notes,

herausgegeben von International Business Machines Corporation, USA.

© Copyright International Business Machines Corporation 2001
© Copyright IBM Deutschland Informationssysteme GmbH 2001

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
Juni 2001

Inhaltsverzeichnis

VisualAge(R) für Java(TM), Professional Edition für Windows(R), Version 4.0 - Release-Hinweise 1

VisualAge(R) für Java(TM), Professional Edition für Windows (R), Version 4.0 - README-Datei 3

Bemerkungen 63

Informationen zu Programmierschnittstellen 67

Marken und Dienstleistungsmarken . . 69

VisualAge(R) für Java(TM), Professional Edition für Windows(R), Version 4.0 - Release-Hinweise

Einführung

Für alle Sprachen können die Release-Hinweise, sas Installations- und Migrationshandbuch sowie das Handbich zur Fehlerbehebung im Hilfesystem auch im Web unter http://www.ibm.com/vadd abgerufen werden.

Release-Hinweise sind komponentenspezifisch und enthalten Informationen über Komponentenvoraussetzungen, Einschränkungen und bekannte Probleme. Darüber hinaus enthalten sie auch allgemeine Tipps und Hinweise zu der jeweiligen Komponente. Allgemeine Informationen bezüglich Voraussetzungen, Migration und Installation von VisualAge für Java finden Sie im Installations- und Migrationshandbuch. Schlagen Sie im Handbuch zur Fehlerbehebung im Hilfesystem nach, wenn Sie Informationen zur Lösung von Problemen im Hilfesystem benötigen.

Allen Benutzern wird dringend empfohlen, den Abschnitt “Wichtige Hinweise für alle Benutzer von VisualAge für Java” zu lesen.

Inhaltsverzeichnis

- Wichtige Hinweise für alle Benutzer von VisualAge für Java (page 1)
- Migration von vorherigen Releases (page 1)
- Fehlerbehebung im Hilfesystem (page 2)
- Hinweise nach Produktkomponenten (page 2)
- Copyrights und Marken

Wichtige Hinweise für alle Benutzer von VisualAge für Java

- README (in HTML-Format)
- Installations- und Migrationshandbuch (Verzeichnis ’instmigr’ auf der CD)
- Hilfeprobleme in IBM Softwareprodukten lösen (Verzeichnis ’Troubleshoot’ auf der CD)
- Erste Schritte
- Websphere Test-Umgebung
- Basiskomponenten
  - Integrated Development Environment (IDE)
  - Visual Composition Editor
  - Java-Compiler und zugehörige Abschnitte
  - Java-Klassenbibliotheken und Programmausführung
  - Online-Hilfe

Migration von vorherigen Releases

- Informationen zur Migration von einem früheren Release finden Sie im Installations- und Migrationshandbuch im Verzeichnis ’instmigr’ der Produkt-CD.

Fehlerbehebung im Hilfesystem

- Schlagen Sie im Handbuch "Hilfeprobleme in IBM Softwareprodukten lösen" (Verzeichnis 'Troubleshoot' auf der CD) nach, wenn Sie Informationen zur Lösung von Problemen im Hilfesystem von VisualAge für Java benötigen.

Hinweise nach Produktkomponenten

- Datenzugriff (Data Access)
  - Data Access Beans für den Zugriff auf Daten
  - SQLJ-Unterstützung
  - DB2(R) Stored Procedure Builder (Erstellungsprogramm für gespeicherte Prozeduren)
- Distributed Debugger
- Externe Versionssteuerung
- JSP/Servlet-Entwicklungsumgebung
- Ferner Zugriff auf die Tool-API
- XML für Java Parser

Copyrights und Marken

(C) Copyright IBM Corp. 1997, 2001. All Rights Reserved.
(C) Copyright Deutschland GmbH 1997, 2001. Alle Rechte vorbehalten.

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Der folgende Abschnitt gilt nicht für Großbritannien und andere Länder, in denen solche Bereitstellungen nicht den gültigen Gesetzen entsprechen:

INTERNATIONAL BUSINESS MACHINES CORPORATION STELLT DIESE VERÖFFENTLICHUNG "UNTER ABBEDINGEN DER HAFTUNG FÜR EINEN BESTIMMTEN ZWECK" ZUR VERFÜGUNG, OHNE GEWÄHRLEISTUNGEN JEGLICHER ART, WEDER AUSDRÜCKLICHEN NOCH IMPLIZIERTEN, EINSCHLIESSLICH DER, ABER NICHT BESCHRÄNKT AUF DIE IMPLIZIERTEN GEWÄHRLEISTUNGEN UND BEDINGUNGEN DER NICHT-VERLETZUNG GEWERBLICHER SCHUTZRECHTE, DER MARKTGÄNGIGEN QUALITÄT UND DER EIGNUNG FÜR EINEN BESTIMMTEN ZWECK. In einigen Ländern sind Ablehnungserklärungen bezüglich ausdrücklicher oder implizierter Gewährleistungen nicht zulässig, daher trifft diese Erklärung möglicherweise nicht für Sie zu. Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen bzw. neuen Editionen der Veröffentlichung bekanntgegeben. IBM kann jederzeit ohne vorherige Ankündigung Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in dieser Veröffentlichung auf Web-Sites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Web-Sites dar.

Das über diese Web-Sites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Web-Sites geschieht auf eigene Verantwortung. Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht. Die Lieferung des im Dokument aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programme oder einer äquivalenten Vereinbarung.

IBM, AIX, AS/400, DB2, OS/390, OS/400, RS/6000, S/390, VisualAge und WebSphere sind Marken der IBM Corporation in den USA und/oder anderen Ländern.

Lotus, Lotus Notes und Domino sind Marken der Lotus Development Corporation in den USA und/oder anderen Ländern. Java und alle Java-basierten Marken und Logos sind Marken der Sun Microsystems, Inc. in den USA und/oder anderen Ländern. Microsoft, Windows und Windows NT sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern. Intel und Pentium sind Marken der Intel Corporation in den USA und/oder anderen Ländern. Andere Unternehmens-, Produkt- und Dienstleistungsnamen können Marken oder Dienstleistungsmarken anderer Unternehmen sein.

VisualAge(R) für Java(TM), Professional Edition für Windows (R), Version 4.0 - README-Datei

Die neueste README-Datei für alle Sprachen finden Sie im Web unter http://www.ibm.com/vadd.

Diese README-Datei enthält die folgenden Informationen zu VisualAge für Java, Professional Edition.

- Bekannte grundsätzliche Probleme und Einschränkungen
- Inhalt der CD

Informationen in allen Sprachen zur Installation von VisualAge für Java entnehmen Sie bitte dem Installations- und Migrationshandbuch (instmigr.htm), das sich im Verzeichnis 'instmigr' auf der Produkt-CD befindet. Falls Sie eine elektronische Version von VisualAge für Java haben, finden Sie diese Handbuchdatei in Ihrem temporären Verzeichnis, in das Sie Ihre Archivpakete entpackt haben. Das neueste Handbuch für alle Sprachen finden Sie im Web unter http://www.ibm.com/vadd.

Informationen in allen Sprachen zur Fehlerbehebung im Hilfesystem finden Sie im Handbuch "Hilfeprobleme in IBM Software-Produkten lösen" (trshoot.htm), das sich im Verzeichnis 'Troubleshoot' auf der Produkt-CD befindet. Falls Sie eine elektronische Version von VisualAge für Java haben, finden Sie diese Handbuchdatei in Ihrem temporären Verzeichnis, in das Sie Ihre Archivpakete entpackt haben. Für alle Sprachen kann die Handbuchdatei im Web unter http://www.ibm.com/vadd abgerufen werden.

Diese Datei enthält keine detaillierten Informationen über die spezifischen Komponenten und Elemente von VisualAge für Java, Professional Edition. Diese Informationen finden Sie in allen Sprachen in den Release-Hinweisen (stehen nach der Installation des Produkts zur Verfügung) zum Produkt. Wählen Sie hierzu **Start -> Programme -> VisualAge für Java für Windows V.4.0 -> Release-Hinweise** aus. Für alle Sprachen können die Release-Hinweise im Web unter http://www.ibm.com/vadd abgerufen werden.

Die Release-Hinweise zu jeder **Komponente** enthalten einen Dokumentationsabschnitt mit einer Liste wichtiger Änderungen in der Online-Information, die nicht übersetzt worden sind.

Einige Online-Hilfedateien liegen als PDF-Dokumente vor. Informationen zum jeweiligen Inhalt der einzelnen PDF-Dateien finden Sie im PDF-Index im Handbuch "Erste Schritte". Nicht für jede Sprache steht auch eine landessprachliche PDF-Version zur Verfügung. Nicht alle PDF-Dateien wurden für Version 4.0 aktualisiert. Die PDF-Dateien stehen im Verzeichnis 'pdf' zur Verfügung. Dieses Verzeichnis befindet sich auf der Produkt-CD. Falls Sie eine elektronische Version von VisualAge für Java haben, finden Sie das Verzeichnis in Ihrem temporären Verzeichnis, in das Sie Ihre Archivpakete entpackt haben. Wenn Sie nicht das Archiv heruntergeladen haben, das die PDF-Dateien enthält, steht dieses Verzeichnis jedoch nicht zur Verfügung.

Inhaltsverzeichnis

1.0 Bekannte grundsätzliche Probleme und Einschränkungen
1.1 DBCS-Probleme
1.2 JDBC-Timing-Fenster
1.3 Probleme beim Systemstart
1.4 Mehrere Bildschirme unter Windows 2000
1.5 Zwei oder mehr Internet-Browser
1.6 Dialog 'Programme hinzufügen/entfernen' schlägt fehl bei der Installation von einem elektronischen Image
1.7 Mauszeigerschatten unter Windows 2000
1.8 Installation von VisualAge für Java, Version 4.0 unter Windows 98 kann eine GPF-Ausnahmebedingung hervorrufen
2.0 Inhalt der CD

1.0 Bekannte grundsätzliche Probleme und Einschränkungen

In diesem Abschnitt werden allgemeine Probleme und Einschränkungen behandelt, die einen Großteil der Benutzer betreffen. Informationen zu Problemen bei der Installation bzw. Deinstallation entnehmen Sie bitte dem Installations- und Migrationshandbuch. Komponentenspezifische Einschränkungen und bekannte Probleme sind in den Release-Hinweisen ausführlicher behandelt.

1.1 DBCS-Probleme

- In den Teilfenstern **Javadoc für Methoden** und **Javadoc für Typen** des Dialogs **Optionen** (wählen Sie **Fenster -> Optionen -> Codierung -> Formatierer -> Javadoc für Methoden** (oder **Javadoc für Typen**) können Sie die Javadoc-Kommentare angeben. Einige DBCS-Zeichen werden in den Kommentaren möglicherweise nicht korrekt angezeigt. Wenn dies auftritt, ändern Sie die beschädigten Zeichen im Teilfenster **Quelle**.

- Der IBM JDK kann die Sprache 'Vereinfachtes Chinesisch' in einigen Dialogen der Sprache 'Traditionelles Chinesisch' anzeigen.
- Alle beanNames verwenden den ASCII-Zeichensatz. Wenn Sie eine Zeichenfolge (string) in eine Eigenschaftsressourcendatei (Property Resource File) unter Verwendung von nicht-ASCII-Zeichen als beanName isolieren, zeigt der Visual Composition Editor (VCE) keinen String für die Bean, und die Bean wird zur Laufzeit nicht angezeigt.

1.2 Durch ein JDBC Timing Hole schließt die Bereinigungsfunktion DB2-Objekte, auf die Referenzen bestehen

Durch die Bereinigungsfunktion werden möglicherweise einige DB2-Objekte geschlossen, auf die in VisualAge für Java interne Referenzen bestehen. Dadurch kann es zu einem Core Dump kommen. Diese Situation tritt überwiegend auf schnellen Maschinen auf, bei denen die Anwendungsprogrammierer zur Bereinigung von JDBC-Objekten auf die Bereinigungsfunktion (Garbage Collector, GC) angewiesen sind.

Die notwendigen Fixpaks können von der folgenden Web-Site heruntergeladen werden:

ftp://ftp.software.ibm.com/ps/products/db2/fixes/

Sie müssen dabei das entsprechende Sprachenverzeichnis und Plattform/Version von DB2 auswählen. DB2 V7.2 ist im Lieferumfang der Produkt-CD von VisualAge für Java für Windows 4.0 enthalten.

1.3 Probleme beim Systemstart

Wenn Sie die folgende Fehlermeldung während des Systemstarts erhalten:

```
Der/die folgende(n) Fehler traten während der Systemstart-Sequenz auf
und
kann/können möglicherweise die korrekte Durchführung beeinträchtigen:

1)Fehler beim Zugriff auf ivj.dat: OS-Error 11001
2)Fehler beim Zugriff auf ivj.dat: OS-Error 11001

Möchten Sie trotzdem fortfahren?
```

Prüfen Sie, ob die Serveradresse in Ihrer ide.ini-Datei aufgelöst und 'gepingt' werden kann. Sie müssen gegebenenfalls Ihren Netzwerkadministrator kontaktieren und bestätigen, dass Ihre Maschine vorschriftsmäßig konfiguriert ist.

1.4 Mehrere Bildschirme unter Windows 2000

VisualAge für Java bietet eingeschränkte Unterstützung für die Verwendung mehrerer Bildschirme unter Windows 2000.

1.5 Zwei oder mehr Internet-Browser

Auf Systemen mit zwei oder mehr Internet-Browsern, können durch den Aufruf der README oder des Installations- und Migrationshandbuch beim Installationsstart mehrere Browser gestartet werden.

1.6 Dialog 'Programme hinzufügen/entfernen' schlägt fehl bei der Installation von einem elektronischen Image

Wenn Sie mit einem elektronischen Image von VisualAge für Java arbeiten, können Sie den Dialog **Hinzufügen/Entfernen** (**Start -> Einstellungen -> Systemsteuerung -> Programme hinzufügen/entfernen**) nicht verwenden, um zusätzliche VisualAge für Java-Komponenten nach der ersten Installation zu installieren. Wenn Sie dies dennoch tun, erhalten Sie eine Fehlernachricht und können keine zusätzlichen Komponenten installieren. Sie müssen setup.exe von dem Pfad aus ausführen, in dem Sie die heruntergeladenen Teile extrahiert haben, damit alle zusätzlichen Komponenten VisualAge für Java hinzugefügt werden können.

1.7 Mauszeigerschatten unter Windows 2000

Benutzerschnittstellen, die mit Swing-Komponenten unter Windows 2000 erstellt wurden, können Probleme beim Aktualisieren haben. Um das Problem zu beheben, inaktivieren Sie die Funktion für die Anzeige des Mauszeigerschattens unter Windows 2000. Rufen Sie das Dialogfenster **Eigenschaften von Maus** (durch Auswahl von **Start**, dann **Systemsteuerung** und anschließend urch Anklicken des Symbol **von Maus**) in Windows 2000auf, klicken Sie die Registerkarte **Zeiger** ans, und inaktivieren Sie das Markierungsfeld **Zeigerschatten aktivieren**.

1.8 Installation von VisualAge für Java, Version 4.0 unter Windows 98 kann eine GPF-Ausnahmebedingung hervorrufen

Wenn Sie VisualAge für Java, Version 4.0 auf Windows 98 installieren, kann folgende Nachricht angezeigt werden: _DELIS caused a general protection fault in module _ISDEL.EXE. (_DELIS verursachte einen allgemeinen Schutzfehler in Modul _ISDEL.EXE.) Dieses Nachricht tritt dann auf, wenn eine neuere Version von msiexec.exe (im Systemverzeichnis) auf Ihrer Workstation als die, die für die Installation von VisualAge fpr Java, Version 4.0, verwendet wurde, vorhanden ist. Klicken Sie **Schließen** im Nachrichtendialog an. Die Installation von VisualAge für Java, Version 4.0, wird anschließend ohne Probleme fortgesetzt.

2.0 CD-Inhalt

- Installationsprogramm für VisualAge für Java
- Distributed Debugger
- Verzeichnis 'Extras'
- IBM Developer Kit, Java Technology Edition, v1.2.2
- Das Verzeichnis 'instmigr' enthält das Installation- und Migrationshandbuch in jeder Sprache.
- Das Verzeichnis 'ivj40\backup' enthält die zur Wiederherstellung erforderlichen Dateien.
- Das Verzeichnis 'pdf' enthält die ausdruckbaren Dokumentationsdateien (im PDF-Format).
- Das Verzeichnis 'README' enthält die Produkt-Readme in jeder Sprache.
- Das Verzeichnis 'Troubleshoot' enthält das Handbuch zur Fehlerbehebung im Hilfesystem in jeder Sprache.
- Liste der Fixes für VisualAge für Java, Version 4.0 (fixlist.txt)

Das Verzeichnis 'Extras' auf der CD enthält folgendes:

- Runtimes für VisualAge für Java, Version 3.0. Der Name des Unterverzeichnisses lautet 'runtime30'.
- Laufzeiten für VisualAge für Java, Version 3.5. Der Name des Unterverzeichnisses lautet 'runtime35'.

Copyright und rechtliche Hinweise

(C) Copyright IBM Corporation 2000, 2001. All Rights Reserved. (C) Copyright IBM Deutschland GmbH 2000, 2001. Alle Rechte vorbehalten.

IBM, AIX, AS/400, DB2, OS/390, OS/400, RS/6000, S/390, VisualAge und WebSphere sind Marken der IBM Corporation in den USA und/oder anderen Ländern. Java und alle Java-basierten Marken und Logos sind Marken der Sun Microsystems, Inc. in den USA und/oder anderen Ländern. Microsoft, Windows und Windows NT sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern. Andere Unternehmens-, Produkt- und Dienstleistungsnamen können Marken oder Dienstleistungsmarken anderer Unternehmen sein.

Der folgende Abschnitt gilt nicht für Großbritannien und andere Länder, in denen solche Bereitstellungen nicht den gültigen Gesetzen entsprechen:

INTERNATIONAL BUSINESS MACHINES CORPORATION STELLT DIESE VERÖFFENTLICHUNG "UNTER ABBEDINGEN DER HAFTUNG FÜR EINEN BESTIMMTEN ZWECK" ZUR VERFÜGUNG, OHNE GEWÄHRLEISTUNGEN JEGLICHER ART, WEDER AUSDRÜCKLICHEN NOCH IMPLIZIERTEN, EINSCHLIESSLICH DER, ABER NICHT BESCHRÄNKT AUF DIE IMPLIZIERTEN GEWÄHRLEISTUNGEN UND BEDINGUNGEN DER NICHT-VERLETZUNG GEWERBLICHER SCHUTZRECHTE, DER MARKTGÄNGIGEN QUALITÄT UND DER EIGNUNG FÜR EINEN BESTIMMTEN ZWECK.

In einigen Ländern sind Ablehnungserklärungen bezüglich ausdrücklicher oder implizierter Gewährleistungen nicht zulässig, daher trifft diese Erklärung möglicherweise nicht für Sie zu. Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen bzw. neuen Editionen der Veröffentlichung bekanntgegeben. IBM kann jederzeit ohne vorherige Ankündigung Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

VisualAge(R) für Java(TM)

Version 4.0

WebSphere(TM) Test Environment

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einführung
2.0 Voraussetzungen
    2.1 Hardware-Voraussetzungen
    2.2 Unterstützung für Entwicklungsplattformen

2.3 Software-Unterstützung
3.0 Einschränkungen und bekannte Probleme
3.1 Fehler beim Neuaufbau des WebSphere-Testumgebungs-Kontrollzentrums unter Windows 2000
3.2 Das WebSphere-Testumgebungs-Kontrollzentrum erscheint beim Neustart nicht im Vordergrund
3.3 JSP-Dateien können nicht in einem Verzeichnis gesichert werden, dessen Name einen "." enthält"."
3.4 DBCS-Zeichen können nicht in einem DataSource-Namen verwendet werden
3.5 Ein DataSource-Objekt mit einem langen Namen erstellen
3.6 DataSource API-Schnittstelle befindet sich nicht in der Laufzeit
3.7 Geänderte DataSource-Attribute werden nicht verwendet
3.8 JSP aktualisiert keine Änderungen
3.9 Probleme beim Zugriff auf den LDAP-Server
3.10 Einschränkungen des Klassenpfads
3.11 Einen Klassenpfadeintrag im Klassenpfad der Servlet-Engine löschen
3.12 Websphere-Testumgebung enthält Code anderer Hersteller
4.0 Dokumentation

1.0 Einführung

Diese Release-Hinweise enthalten Informationen über verschiedene Einschränkungen und bekannte Probleme bei der WebSphere-Testumgebung. Dieses Dokument enthält außerdem Informationen zu Hardware- und Software-Voraussetzungen sowie Hinweise, Tipps und Lösungen zur Problemumgehung.

2.0 Voraussetzungen

2.1 Hardware-Voraussetzungen

Die Hardware-Voraussetzungen für den Client sind dieselben wie für IBM(R) VisualAge für Java, Enterprise Edition für Windows, Version 4.0 angegebenen, mit folgenden Ausnahmen:

- Ein Pentium (R) II-Prozessor oder höher wird empfohlen.
- Ein Prozessor mit einer Minimalgeschwindigkeit von 400 MHz wird empfohlen.
- Es sind mindestens 128 MB Arbeitsspeicher (RAM) erforderlich. Zur Vermeidung einer Plattenüberlastung werden jedoch 256 MB RAM empfohlen.

Wenn Sie den WebSphere Application Server mit DB2(R) und VisualAge für Java gleichzeitig ausführen möchten, wird ein Minimum von 512 MB empfohlen.

2.2 Unterstützung für Entwicklungsplattformen

Die WebSphere-Testumgebung unterstützt das Testen auf folgenden Entwicklungsplattformen: Windows NT(R), Version 4, und Windows 2000.

2.3 Software-Unterstützung

- Die WebSphere-Testumgebung unterstützt folgende Objekte:
  - JSP 0.91-Spezifikationen
  - JSP 1.0-Spezifikationen
  - JSP 1.1-Spezifikationen
  - Servlet 2.1-Spezifikationen
  - Servlet 2.2-Spezifikationen

  - DataDirect-SequeLink 5.1-JDBC-Treiber (JDBC: Java Database Connectivity) von Merant für Microsoft SQL Server und Oracle-Datenbankzugriff. (Weitere Informationen zum Installieren und Konfigurieren von Merant SequeLink Server enthält das *Installations- und Migrationshandbuch*.)
- Die JTA-Unterstützung erfordert DB2 6.1 Fix Pack 4 und höher oder DB2 7.1.
- Diese DB2-Stufen unterstützen die Java Transaction APIs (JTA) über den DB2 JDBC-Anwendungstreiber. DB2 stellt keine JTA-Unterstützung mit dem DB2 JDBC-Netztreiber zur Verfügung.
- Die WebSphere-Testumgebung bietet *keine* Unterstützung für folgende Objekte:
  - shtml
  - ORBs (Object Request Brokers) anderer Hersteller, z. B. Visibroker ORB
  - JTA-fähige Sybase-DataSources

3.0 Einschränkungen und bekannte Probleme

3.1 Fehler beim Neuaufbau des WebSphere-Testumgebungs-Kontrollzentrums unter Windows Windows(R)

Wenn Sie VisualAge für Java auf Windows 2000 verwenden, kann es sein, dass einige der GUIs (grafischen Benutzeroberflächen) nicht korrekt neu aufgebaut werden. Minimieren Sie die GUI und maximieren Sie sie anschließend erneut, um dieses Problem zu umgehen. Die GUI wird dadurch ordnungsgemäß wiederhergestellt.

3.2 Das WebSphere-Testumgebungs-Kontrollzentrum erscheint beim Neustart nicht im Vordergrund

Das WebSphere-Testumgebungs-Kontrollzentrum erscheint nicht im Vordergrund, wenn Sie es erneut starten. Wählen Sie das Symbol **Kontrollzentrum** aus der Symbolleiste aus, um das Kontrollzentrum wieder in den Vordergrund zu bringen.

3.3 3.3 JSP-Dateien können nicht in einem Verzeichnis gesichert werden, dessen Name einen "." enthält

Wenn Sie eine JSP-Datei in einem Verzeichnis sichern, dessen Name einen "." enthält (z. B. JSP1.0), zeigt der Browser einen Kompilationsfehler an. Dies ist eine Einschränkung der WebSphere-Testumgebung.

3.4 DBCS-Zeichen können nicht in einem DataSource-Namen verwendet werden

Sie können in einem DataSource-Namen keine DBCS-Zeichen (Double-Byte Character Set) verwenden. Wenn Sie versuchen, ein DataSource-Objekt hinzuzufügen, dessen Name ein DBCS-Zeichen enthält, so erscheint die folgende Fehlermeldung:

```
javax.naming.NamingException: root exception is
org.omg.CORBA.DATA_CONVERSION minor code:6 completed: no
```

3.5 Ein DataSource-Objekt mit einem langen Namen erstellen

In manchen Fällen wird beim Erstellen eines DataSource-Objekts mit einem sehr langen Namen der Eintrag im Teilfester für die DataSource-Liste nicht sofort korrekt angezeigt. Dies können Sie beheben, indem Sie das Fenster vergrößern, damit der Name vollständig angezeigt werden kann.

3.6 Die DataSource API-Schnittstelle befindet sich nicht in der Laufzeit

Die 'com.ibm.db2.jdbc.app.stdext.javax.sql.DataSource'-Schnittstelle ist nicht länger Bestandteil der WebSphere-Laufzeit. Beim Migrieren von bestehendem Code, der sich auf DataSources bezieht, müssen Sie alle Verweise auf `com.ibm.db2.jdbc.app.stdext.javax.sql.DataSource` in `javax.sql.DataSource` umändern.

3.7 Geänderte DataSource-Attribute werden nicht verwendet

Wenn Sie die Attribute einer DataSource (z. B. den Datenbanktreiber) verändern möchten und Sie das WebSphere-Testumgebungs-Kontrollzentrum zwischen dem Löschen der DataSource und ihrer Wiederherstellung nicht ordnungsgemäß schließen, kann es vorkommen, dass die alten anstelle der neuen Attribute verwendet werden, wenn Sie die DataSource wiederherstellen. Führen Sie folgende Schritte aus, um dieses Problem zu verhindern:

1. Löschen Sie die DataSource auf der DataSource-Konfigurationsseite im WTE-Kontrollzentrum.
2. Fahren Sie den Persistent Namens-Server und das WTE-Kontrollzentrum herunter.
3. Führen Sie einen Neustart des WTE-Kontrollzentrums und des Persistent Namens-Servers durch.
4. Erstellen Sie die DataSource mit den neuen Parametern.

3.8 JSP aktualisiert keine Änderungen

Wenn JSP keine Änderungen aktualisiert, gehen Sie folgendermaßen vor:

1. Bei **JSP 0.91**: Löschen Sie alle im Cache zwischengespeicherten Dateien in den folgenden beiden Verzeichnissen:

   X:\VisualAge for Java\ide\project_resources\IBM WebSphere Test Environment\temp\default_app\pagecompile_jsp
   X:\VisualAge for Java\ide\project_resources\IBM WebSphere Test Environment\temp\default_app\pagecompile\etc_jsp

   (Hierbei ist *X:\VisualAge for Java* das Verzeichnis, in dem VisualAge für Java installiert wurde.

   Bei **JSP 1.0**: Löschen Sie alle im Cache zwischengespeicherten Dateien im Verzeichnis *X:\VisualAge for Java*\ide\project_resources\IBM WebSphere Test Environment\temp\JSP1_0\default_app\ *JSP-pfadname* (*JSP-pfadname* ist das Verzeichnis, das die JSP-Dateien enthält).

   Bei **JSP 1.1**: Löschen Sie alle im Cache zwischengespeicherten Dateien im Verzeichnis *X:\VisualAge for Java*\ide\project_resources\IBM WebSphere Test Environment\temp\JSP1_1\default_app\ *JSP-pfadname*.
2. Löschen Sie alle im Cache zwischengespeicherten Dateien im Verzeichnis *X:\VisualAge for Java*\ide\project_resources\IBM WebSphere Test Environment\temp\JSP1_0\default_app\etc\ *JSP-pfadname*.
3. Löschen Sie den Cache-Inhalt des Web-Browsers.
4. Starten Sie WTE erneut.
5. Laden Sei die Klassen extern.

3.9 Probleme beim Zugriff auf den LDAP-Server

Wenn Ihre Anwendung nicht auf den LDAP-Server (Lightweight Directory Access Protocol) unter VisualAge für Java zugreifen kann und die Nachricht "Noclassdeffounderror" empfangen wird, gehen Sie folgendermaßen vor:

1. Löschen Sie WTE aus dem Projektpfad der Anwendung.
2. Fügen Sie WTE erneut dem Projektpfad der Anwendung hinzu.

3.10 Einschränkungen des Klassenpfads

Nach dem Starten des Kontrollzentrums der WebSphere-Testumgebung werden sämtliche, vom Benutzer an den Klassenpfadeinstellungen des Arbeitsbereichs (über dessen Auswahlmenü) vorgenommenen Änderungen nicht vom Kontrollzentrum der WebSphere-Testumgebung erkannt, bis dieses geschlossen und erneut gestartet wird.

3.11 Einen Klassenpfadeintrag im Klassenpfad der Servlet-Engine löschen

Wenn Sie einen Klassenpfadeintrag in den Klassenpfadeinstellungen der Servlet-Engine löschen, müssen Sie dieServlet-Engine anschließend manuell stoppen und erneut starten, damit die Änderungen erkannt werden. Ein Neustart der Servlet-Engine funktioniert hier nicht. Beachten Sie, dass dies beim Hinzufügen von Klassenpfadeinträgen zu den Klassenpfadeinstellungen der Servlet-Engine nicht erforderlich ist.

3.12 Websphere-Testumgebung enthält Code anderer Hersteller

Die Websphere-Testumgebung enthält einigen Code anderer Hersteller, einschließlich Merant SequeLink-Treiber (Copyright MERANT, 1991-1999). Ihre Verwendung der Merant SequeLink-Treiber ist auf die Verwendung zum Zweck des Auswertens, Demonstrierens, Entwickelns oder Testens von Anwendungen in der Websphere-Testumgebung.

4.0 Dokumentation

Alle Dateien der WebSphere-Testumgebung wurden seit der letzten Übersetzung geändert. Zusätzlich wurden die folgenden neuen Seiten hinzugefügt:

- "Die Attribute eines DataSource-Objekts ändern" (Changing the attributes of a DataSource object)
- "Die Standard-HTML-Seite ändern" (Changing the default HTML page)
- "Anmerkungen zur Verwendung von DataSource-Objekten" (Considerations for using DataSource objects)
- "Hinweise und Tipps" (Hints and tips)
- "Abrufen von DataSource-Objekten" (Obtaining DataSource objects)
- "Unter JDK entwickelte Servlets ausführen" (Running servlets developed with JDK)

Die Seite “WebSphere-Testumgebungs-Services starten, stoppen und konfigurieren” (Starting, stopping, and configuring WebSphere Test Environment services) enthält einen neuen Abschnitt namens “Systemeigenschaften für die Servlet-Engine definieren” (Setting system properties for the servlet engine).

Die Seite “DataSources mit einem Programm erstellen” (Creating DataSources programmatically) wurde neu strukturiert und überarbeitet.

Ignorieren Sie auf der Seite “Anmerkungen zur Verwendung von DataSource-Objekten” (Considerations for using DataSource objects) den letzten Satz: “Eher anfällig für Fehler als die Verwendung des Kontrollzentrums derWebSphere-Testumgebung”.

Die jeweils neuesten Informationen können Sie der englischen Version des Portable Document Format (PDF) für die WebSphere-Testumgebung entnehmen.

VisualAge(R) für Java(TM)

Version 4.0

Buch ’Erste Schritte’

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Dokumentation

1.0 Dokumentation

Es folgt eine Liste der Abschnitte im Handbuch ″Erste Schritte″, die nicht übersetzt worden sind. Alle aufgeführten Informationen stehen in der englischen PDF-Datei ″Getting Started Guide″ zur Verfügung, die im Verzeichnis “\pdf\en_us” auf der Produkt-CD zu finden ist.

1. ″VisualAge für Java installieren″ (Installing VisualAge for Java)
2. ″Einen Clienten mit todoserv verbinden″ (Connecting a client to the todoserv)
3. ″Optimierung und Debugging″ (Optimization) and Debugging
4. ″Benutzer zum Team-Repository der To-Do-Liste hinzufügen″ (Adding users to the To-Do List team repository)
5. Verbinden der Beans für “ToDoList” (Connecting the To-Do List beans)
6. Aktivieren der Statusprüfung im Applet “ToDoList” (Enabling state chekcing of the To-Do List applet)
7. Starten und Stoppen des Servers (Starting and stopping the server)
8. Zugehörige Informationen (Related information)
9. PDF-Index (PDF index)
10. ″Web-Ressourcen″ (Web resources)
11. Einrichten des Servers für das Team “ToDoList” (Setting up the server for the To-Do List team)
12. Der Enterprise Toolkit für AS/400 (ET/400) (The Enterprise Toolkit for AS/400 (ET/400))
13. ″AS/400 Toolbox for Java″ (AS/400 Toolbox for Java)

14. ″Enterprise Access Builder für Transaktionen″ (Enterprise Access Builder for Transactions)
15. Zugriff auf Transaktions-Server″ (Access to transaction servers)
16. ″Datenbankzugriff″ (Database access)
17. Access Builders
18. ″Weitere Tools″ (Other tools)
19. ″Änderungen in diesem Release von VisualAge für Java″ (What's new in this release of VisualAge for Java)

Die Lokation des Buchs zur Server-Einrichtung und Administration entnehmen Sie bitte der Produkt-Informationsdatei (README).

VisualAge(R) für Java(TM)

Version 4.0

Integrierte Entwicklungsumgebung (IDE)

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einführung
2.0 Einschränkungen und bekannte Probleme
- 2.1 Applets ausführen
  - 2.1.1 In einem Web-Browser
  - 2.1.2 NoClassDefFoundError tritt auf
- 2.2 Debugger - Sicherheitsmanager
- 2.3 Japanische Schriftarten im Dialog 'Optionen'
- 2.4 Bidirektionaler Text
  - 2.4.1 Rückgängig/Wiederherstellen in bidirektionalen Sprachen
  - 2.4.2 Unterstützung für bidirektionale Texte inaktivieren
- 2.5 Sun BDK - Beispiele
  - 2.5.1 Beans zu BeanBox Demo von Sun hinzufügen
  - 2.5.2 BridgeTester-Bean
- 2.6 Sun BDK - Beispiele
  - 2.6.1 Sun JDK Animator
  - 2.6.2 Sun JDK Clock
  - 2.6.3 Sun JDK DrawTest
  - 2.6.4 Sun JDK SymbolTest
- 2.7 AWT-Beispiele
  - 2.7.1 AwtMenuUseCase
- 2.8 Große JTables drucken
- 2.9 Die JDK-Stufe ändern
- 2.10 Eine Methode kopieren, die keinen Quellcode enthält
- 2.11 Fehler 33 tritt beim Starten von VisualAge für Java über einen Direktaufruf auf
- 2.12 SmartGuide ″Applet erstellen″ generiert Methode main() nicht richtig
- 2.13 Externen Quellcode kompilieren
- 2.14 Pakete mit undefinierten Typen importieren
- 2.15 Aventail Autosocks 2.42E ausführen
- 2.16 Quellensicht
  - 2.16.1 Eine Klasse auf zwei verschiedenen Seiten bearbeiten
  - 2.16.2 Kommentare und das Quellensichtsfenster
  - 2.16.3 Ändern der Klassendefinition in der Quellensicht in Work-

bench-Rebuilds
2.17 Seiten in Java2DemoApplet ändern
2.18 Umbenannte Ressourcen nicht immer aktualisiert, wenn ein Projekt erneut geladen wird
2.19 Debuggen von Class-Loaders
2.20 Eine Klasse mit $-Symbolen erstellen
3.0 Hinweise und Tipps
3.1 Arbeitsbereichsprobleme und -fehler behandeln
3.2 Repositoryprobleme und -fehler behandeln
4.0 Dokumentation
4.1 Aktualisierungen der englischen Dokumentation
4.1.1 Task-Datei - Code implementieren
4.2 Unterbrochene Verknüpfungen im Navigationsteilfenster
4.3 Nicht übersetzte Änderungen an der Dokumentation

1.0 Einführung

Diese Release-Hinweise enthalten Informationen über verschiedene Einschränkungen und bekannte Probleme bei der IDE sowie Hinweise, Tipps und Lösungen zur Problemumgehung.

2.0 Einschränkungen und bekannte Probleme

2.1 Applets ausführen

2.1.1 In einem Web-Browser

Wenn Ihr exportiertes Applet im Web-Browser nicht ausgeführt wird, stellen Sie sicher, ob der Browser J2DK 1.1.7 unterstützt. Informieren Sie sich auf der Site des Web-Browser-Lieferanten über Berichtigungen bezüglich JDK. Stellen Sie darüber hinaus sicher, dass die Sicherheitseinstellungen des Web-Browsers zum Ausführen des Applets richtig sind.

2.1.2 NoClassDefFoundError tritt auf

Wenn Sie ein Applet ausführen wollen und diesen Fehler erhalten, müssen Sie die Applet Viewer-Eigenschaften ändern. Gehen Sie dazu folgendermaßen vor:

1. Führen Sie das Applet aus.
2. Wählen Sie nach dem Fehlschlagen der Ausführung **Applet > Eigenschaften** im Applet Viewer aus.
3. Wählen Sie aus dem Dropdown-Menü **Klassenzugriff** die Option **Uneingeschränkt** aus. Klicken Sie **Anwenden** an.
4. Schließen Sie den Applet Viewer und starten Sie das Applet erneut.

2.2 Debugger - Sicherheitsmanager

Das Installieren eines Sicherheitsmanagers, der keinen vollständigen Zugriff auf Member und Eigenschaften erlaubt, kann aufgrund von Introspektionsfehlern zu Ausnahmebedingungen oder fehlenden Feldern im Debugger führen. Dies betrifft den SecurityManager und dessen Unterklassen.

Umgehung des Problems: Verwenden Sie beim Debuggen von Code keinen Sicherheitsmanager oder stellen Sie sicher, dass der Sicherheitsmanager den Zugriff auf Member und Eigenschaften ermöglicht.

2.3 Japanische Schriftarten im Dialog 'Optionen'

Wenn die Dialogschriftart im Dialog 'Optionen' auf einer japanischen Maschine auf eine englische Schriftart gesetzt ist (z. B. Adobe Helvetica), enthält die Schriftart nicht die notwendigen japanischen Zeichen. Aus diesem Grund wird der Dialog 'Optionen' selbst unlesbar.

Umgehung des Problems: Der Dialog 'Optionen' kann dennoch verwendet werden, um die Dialogschriftart wieder auf eine japanische Schriftart zu setzen. Wenn sich der Dialog 'Optionen' in diesem Zustand befindet, müssen anstatt der Maus die Pfeiltasten auf der Tastatur verwendet werden, um eine Auswahl zu treffen.

2.4 Bidirektionaler Text

2.4.1 Rückgängig/Wiederherstellen in bidirektionalen Sprachen
Die Operationen 'Rückgängig' (Undo) und 'Wiederherstellen' (Redo) in den Editoren stehen für bidirektionale Betriebssysteme nicht zur Verfügung.

2.4.2 Unterstützung für bidirektionale Texte inaktivieren
In bidirektionalen Betriebssystemen besteht die Möglichkeit, die bidirektionale Unterstützung in den Texteditoren auszuschalten. Wenn die Unterstützung ausgeschaltet ist, sind die Operationen 'Rückgängig' und 'Wiederherstellen' in den Texteditoren verfügbar.

Um die Unterstützung bidirektionaler Texte zu inaktivieren, fügen Sie dem Abschnitt [Landessprachenunterstützung] der Datei ide.ini im Verzeichnis ide/program folgende Zeile hinzu:

```
EnableBidiSource=false
```

2.5 Sun BDK - Beispiele

2.5.1 Beans zu BeanBox Demo von Sun hinzufügen
Wenn Sie Beans zur BeanBox von Sun hinzufügen möchten, müssen Sie die *.jar-Dateien für die Beans in das folgende Verzeichnis kopieren:

\<ibmvjava>\ide\project_resources\jars

Hierbei ist <ibmvjava> das Verzeichnis, in dem VisualAge für Java installiert ist.

Darüber hinaus müssen Sie die *.jar-Dateien in IBM VisualAge für Java importieren, bevor Sie versuchen, die neuen Beans von der BeanBox von Sun auszuführen. Detaillierte Anweisungen zum Importieren von *.jar befinden sich in der Online-Dokumentation von IBM VisualAge für Java.

2.5.2 BridgeTester-Bean
Die BridgeTester-Bean, die mit Sun BDK mitgeliefert wird, ist nicht im Lieferumfang enthalten. Diese Bean weist einen Fehler im java.beans-Paket von Sun auf.

2.5.3 SUN BDK-Beispiele und javax.infobus
Sie können diese beiden Features nicht gleichzeitig in die DE laden, da die SUN(TM) BDK-Beispiele ebenfalls ein Paket des Namens javax.infobus enthalten. Wenn Sie eines dieser Features in den Arbeitsbereich geladen haben, müssen Sie es zuerst entfernen, bevor Sie das andere Feature laden können.

2.6 Sun BDK - Beispiele

2.6.1 Sun JDK Animator
Wenn Sie während der Ausführung des Animators die Option zum Wiederanlauf im Menü des Applet-Viewers auswählen, werden auf der Konsole Fehler angezeigt. Dasselbe Verhalten zeigt sich, wenn dieses Beispiel unter Sun J2SDK Version 1.2.2 ausgeführt wird.

2.6.2 Sun JDK Clock
Die grafische Zeitanzeige zeigt die aktuelle Systemzeit an, die Zeitangabe in Textform ist jedoch falsch.

2.6.3 Sun JDK DrawTest
Wenn Sie IBM VisualAge für Java unter Windows 98/2000 NT ausführen und 256 Farben verwenden, zeichnet der vierte Radioknopf von links nicht mit der angezeigten Farbe. Dies ist ein bekanntes Problem, das auf die Farbenzuordnung von Windows zurückzuführen ist. Das Problem tritt nicht auf, wenn Sie eine höhere Farbauflösung verwenden.

2.6.4 Sun JDK SymbolTest
Die Symbole werden nicht angezeigt. Statt dessen werden die Escape-Codes für die Symbole angezeigt. Dies ist ein bekanntes Problem.

2.7 AWT-Beispiele

2.7.1 AwtMenuUseCase
Die Option **Become Help Menu** (sowohl für Menü 1 als auch für Menü 2) in diesem Beispiel funktioniert nicht. Wenn Sie **Menü 1 -> Become Help Menu** und anschließend **Menü 2 -> Become Help Menu** auswählen, wird Menü 1 nicht mehr angezeigt. Klicken Sie nicht in den leeren Bereich, in dem sich Menü 1 befand, da die IDE sonst abstürzt. Wenn Sie 'Menu 2 -> Become Help Menu' und anschließend 'Menu 1 -> Become Help Menu' auswählen, passiert dasselbe, nur dass in diesem Fall Menü 2 nicht mehr angezeigt wird.

2.8 Große JTables drucken

Wenn beim Drucken einer großen JTable Probleme auftreten, teilen Sie die Tabelle in kleinere Teile auf und drucken diese einzeln.

2.9 Die JDK-Stufe ändern

VisualAge für Java, Version 4.0, unterstützt Version 1.2.2 des JDK (J2SDK). Sie können die von VisualAge für Java unterstützte JDK-Stufe nicht ändern.

2.10 Eine Methode kopieren, die keinen Quellcode enthält

Eine Methode (ohne Quelle) kann nicht aus einer Klasse in eine andere Klasse kopiert werden, die Sie (mit Quelle) erstellt haben. Wenn Sie dies versuchen, wird möglicherweise eine ähnliche Fehlernachricht wie folgende angezeigt: "<Name Ihrer Klasse> kann nicht geändert werden, da sie aus einer .class-Datei importiert wurde."

2.11 Fehler 33 tritt beim Starten von VisualAge für Java über einen Direktaufruf auf

Wenn Sie versuchen, VisualAge für Java über ein Direktaufrufsymbol auf Ihrem Desktop zu starten und die Starteigenschaften für den Direktaufruf nicht korrekt definiert sind, erhalten Sie möglicherweise eine Nachricht mit “Fehler 33”. Führen Sie folgende Schritte aus, um zu überprüfen, ob die Starteigenschaften korrekt definiert sind:

1. Klicken Sie mit der rechten Maustaste auf das Symbol. Wählen Sie aus dem Kontextmenü die Option **Eigenschaften** aus.
2. Klicken Sie die Indexzunge **Direktaufruf** an.
3. Stellen Sie sicher, dass im Feld **Starten in** der folgende Wert steht: *D:\ibmvjava*\ide\program\, wobei *D:\ibmvjava* das Produktinstallationsverzeichnis ist.

2.12 SmartGuide "Applet erstellen" generiert Methode main() nicht richtig

Wenn Sie ein Applet mit Hilfe des SmartGuides ’CreateApplet’ erstellen und dabei die Option “Ja, ein Applet erstellen, das eigenständig oder in einem Applet Viewer ausgeführt werden kann” auf der Seite ’Applet-Eigenschaften’ im SmartGuide auswählen, funktioniert die generierte Methode main() nicht einwandfrei.

Versuchen Sie EINE der folgenden Optionen zum Beheben dieses Problems:

1. Löschen Sie die Methode main() nach dem Erstellen des Applets, und öffnen Sie das Applet anschließend im Visual Composition Editor. Wählen Sie dort **Bean > Code erneut generieren** aus. Die Methode main() wird daraufhin erneut generiert und funktioniert anschließend einwandfrei.
2. Darüber hinaus können Sie die Methode main() in einem Quellcodeteilfenster manuell editieren. Hierzu müssen Sie die Methode frame.show hinter die Anweisung verschieben, die das Applet initialisiert (applet.init).
3. Wählen Sie beim Erstellen des Applets die Option “Nein, ein Applet erstellen, das nur in einem Applet Viewer ausgeführt werden kann” aus. Öffnen Sie das Applet im Visual Composition Editor und wählen Sie **Bean > Code erneut generieren** aus. Die Methode main() wird daraufhin generiert und funktioniert einwandfrei.

2.13 Externen Quellcode kompilieren

VisualAge für Java kann keinen Quellcode kompilieren, der für die IDE extern ist (d. h. der nicht in den Arbeitsbereich importiert wurde sondern nur im Dateisystem vorhanden ist).

2.14 Pakete mit undefinierten Typen in das Repository von VisualAge für Java, Professional Edition importieren

Wenn Sie ein Paket in VisualAge für Java, Enterprise Edition erstellen, das undefinierte Typen enthält, und dieses Paket anschließend versionieren und in ein Repository exportieren, dürfen Sie dieses Paket nicht in ein Repository von VisualAge für Java, Professional Edition importieren, da Sie keine neue Edition davon erstellen können.

Undefinierte Typen werden in VisualAge für Java, Professional Edition nicht unterstützt, da keine Möglichkeit gegeben ist, sie inVisualAge für Java, Professional Edition zu erstellen. Wenn Sie daher versuchen, eine neue Edition eines Pakets zu erstellen, das undefinierte Typen enthält, schlägt die Operation fehl.

Weitere Informationen über undefinierte Typen enthält die IDE-Onlinehilfefunktion von VisualAge für Java.

2.15 Aventail Autosocks 2.42E und VisualAge für Java gleichzeitig ausführen

Wenn Sie VisualAge für Java auf einer Maschine installieren und ausführen, auf der auch Aventail Autosocks (ein Socks-Client, der Netzanforderungen an einen oder mehrere Socks-Server überträgt) Version 2.42E ausgeführt wird, schlagen einige IDE-Operationen (z. B. das Erstellen von Projekten und Paketen) möglicherweise fehl. Es empfiehlt sich daher dringend, *keine* Version von Aventail Autosocks und VisualAge für Java gleichzeitig auf derselben Maschine auszuführen.

2.16 Quellensichtsfenster

2.16.1 Eine Klasse auf zwei verschiedenen Seiten im Quellensichtsfenster editieren

Wenn im Quellensichtsfenster zwei Seiten für dieselbe Klasse geöffnet sind, und Sie Änderungen auf beiden Seiten vornehmen, werden nur die Änderungen auf der Seite beibehalten, die zuerst gesichert wurde. Die Änderungen auf der anderen (nicht gesicherten) Seite gehen verloren. Die nicht gesicherte Seite wird mit den Änderungen von der gesicherten Seite aktualisiert.

2.16.2 Kommentare und das Quellensichtsfenster

In VisualAge für Java, Version 3.5 gehen beim Sichern von Quellcode im Quellensichtsfenster oder beim Importieren des Codes in die IDE Kommentare, die sich vor der Paketdeklaration oder zwischen der Paket- und Typdeklaration befinden, gelegentlich verloren oder werden dupliziert, wenn der entstehende Typ im Quellensichtsfenster angezeigt oder in eine Quellendatei exportiert wird.

Dieses Problem tritt auf, wenn Kommentare vor der Paketdeklaration und Kommentare zwischen der Paket- und der Typendeklaration “kombiniert” wurden. Wenn ein Typ angezeigt wird, der Kommentare vor der Paketdeklaration und Kommentare zwischen der Paket- und der Typendeklaration in der Workbench enthält, werden diese Kommentare “kombiniert”, da die Paketdeklaration in der Workbench nicht angezeigt wird. Wenn Sie beispielsweise einen Typ mit Paketkommentaren in der Workbench anzeigen, erscheint er folgendermaßen:

```
//Package comments
//Type comments
```

Wenn Sie den Typ jedoch im Quellensichtsfenster angezeigt, erscheint er folgendermaßen:

```
//Package comments
package mypackage
//Type comments
```

Die Workbench kann den kombinierten Kommentar richtig anzeigen, er wird jedoch vom Quellensichtsfenster nicht richtig behandelt und auch dann nicht, wenn der Typ in eine Quellendatei exportiert wird. Wenn Sie daher einen Typ mit einem kombinierten Kommentar in der Workbench anzeigen und ihn anschließend im Quellensichtsfenster anzeigen oder exportieren, werden Kommentare im Typ möglicherweise nicht richtig angezeigt.

In VisualAge für Java, Version 4.0 werden Kommentare vor einer Paketdeklaration stets beibehalten. Sie können jedoch nur im Quellensichtsfenster angezeigt werden oder wenn der Typ, der die Kommentare enthält, in eine Quellendatei exportiert wird. Kommentare zwischen der Paket- und der Typendeklaration können in einem beliebigen Quellcode-Teilfenster angezeigt werden. Typen, die im Quellensichtsfenster gesichert oder in die IDE in Version 3.5 importiert wurden und den Fehler von duplizierten oder fehlenden Kommentaren aufweisen, müssen im Quellensichtsfenster der Version 4.0 erneut gesichert bzw. erneut importiert werden.

2.16.3 Ändern der Klassendefinition in der Quellensicht in Workbench-Rebuilds

Die Quellensicht ist für Entwickler konzipiert, die im vollständigen Quellen(datei-)modus arbeiten wollen, und nicht für Entwickler, die in einem gemischten Modus von Dateien und Methoden gleichzeitig arbeiten wollen. Wenn daher eine Klassendefinition in ein Nicht-Quellensichtfenster geändert wird (wenn Sie beispielsweise eine Klasse in der Workbench auswählen, und ein Feld hinzufügen), wird die Quellensicht erneut aufgebaut. Die Klassenmethoden werden innerhalb der Quellensicht und beim Exportieren des Java-Codes alphabetisch umgeordnet.

2.17 Seiten in Java2DemoApplet ändern

Wenn Sie Java2DemoApplet mit '.' im Klassenpfad ausführen, und versuchen, darin Seiten zu ändern, wird eine java.security.AccessControlException auftreten. Sie müssen sicherstellen, dass der Klassenpfad nicht '.' enthält, bevor Sie Java2DemoApplet ausführen, wenn Sie in der Lage sein wollen, Seiten darin zu ändern.

2.18 Umbenannte Ressourcen nicht immer aktualisiert, wenn ein Projekt erneut geladen wird

Wenn Sie ein Projekt in den Arbeitsbereich erneut laden, das eine Ressource enthält, die seit dem letzten Laden des Projekts in den Arbeitsbereich umbenannt wurde, wird die ursprüngliche Ressource nicht aus der Projektressourcenseite oder dem lokalen Ressourcenverzeichnis des Projekts entfernt. Der Arbeitsbereich enthält sowohl die Ressource mit dem ursprünglichen Namen als auch die Ressource mit dem neuen Namen. Die Ressource mit dem ursprünglichen Namen wird als "Fehlend ("Missing") auf der Registerkarte **Projektressourcen** gekennzeichnet und kann nicht gelöscht werden, da sie nicht länger Teil des Projekts ist (dies bedeutet: eine Ressource mit diesem Namen ist nicht mehr im Projekt vorhanden, da die Ressource umbenannt wurde).

Problemumgehung: Löschen Sie das Projekt aus dem Arbeitsbereich und fügen Sie es anschließend dem Arbeitsbereich wieder hinzu.

2.19 Debuggen von Class-Loaders

Das Debugging von Klassen-Loader kann ein gegenseitiges Sperren in gewissen Situationen hervorrufen. Dies kann sich in entweder in einer ClassNotFoundException auswirken, oder es wird ein Dialog aufgerufen (″Die ausgewählte Bean oder der Benutzer-Java-Code antwortet nicht, ″The selected bean or user java code is not responding″). Dieses Problem tritt in der Regel dann auf, wenn mit Step-Into-Operationen in Klassen-Loader eingetreten wird oder wenn Variablen der Klasse untersucht werden, die in den Klassen-Loader geladen werden.

2.20 Eine Klasse mit $-Symbolen erstellen

Beim Erstellen einer Klasse in der IDE, die mit einem $-Symbol beginnt, müssen Sie möglicherweise die Anzeige aktualisieren, um die Klasse sehen zu können.

3.0 Hinweise und Tipps

3.1 Arbeitsbereichsprobleme und -fehler behandeln

In der folgenden Tabelle werden einige Arbeitsbereichsprobleme und -fehler aufgeführt, die auftreten können. Die IDE-Onlinehilfefunktion oder die IDE PDF im Verzeichnis 'pdf' auf der Produkt-CD enthalten weitere Informationen zur Ausführung der in der Spalte 'Lösung' unten aufgeführten Tasks.

Problem	**Lösung**
Nach dem Wechsel zwischen Repositories treten Arbeitsbereichsprobleme auf	Entfernen Sie alle Projekte aus dem Arbeitsbereich, die im alten Repository enthalten waren, aber nicht im neuen enthalten sind.
Bei jedem IDE-Start wird der Arbeitsbereich wiederhergestellt. Dabei spielt es keine Rolle, auf welches Repository zugegriffen werden soll.	Stellen Sie sicher, dass auf die Arbeitsbereichsdatei (ide.icx) nicht nur Lesezugriff besteht. Besteht nicht nur Lesezugriff, dann ist die Datei möglicherweise beschädigt. Um dies zu beheben, müssen Sie den Arbeitsplatz mit Hilfe einer Sicherungskopie ersetzen oder - falls Sie keine Sicherungskopie haben - den mit VisualAge für Java gelieferten Arbeitsbereich erneut installieren.
Wenn Sie versuchen, Code aus dem Repository in den Arbeitsbereich zu laden, stürzt der Arbeitsbereich ab.	Möglicherweise ist Ihr Repository beschädigt. Installieren Sie ein neues Repository und versuchen Sie, den Code aus dem beschädigten Repository in das neue Repository zu importieren.
Beim Öffnen oder Schließen der IDE treten mehrere Basisstörungen oder andere Fehler auf.	Stellen Sie sicher, dass auf die Arbeitsbereichsdatei (ide.icx) nicht nur Lesezugriff besteht. VisualAge für Java sichert beim Öffnen oder Schließen der IDE einige Klassen im lokalen Arbeitsbereich, und dies ist ohne Schreibzugriff nicht möglich.

Beim Starten von VisualAge für Java erhalten Sie eine ähnliche Fehlernachricht wie die folgende: VisualAge kann aufgrund eines nicht behebbaren Fehlers in der Startreihenfolge nicht gestartet werden. Möglicherweise fehlen eine oder mehrere erforderliche Dateien, oder der Arbeitsbereich ist beschädigt.	Wenn Sie versuchen, eine Verbindung zu einem gemeinsam benutzten Repository herzustellen, überprüfen Sie, ob EMSRV aktiv ist. Versuchen Sie, VisualAge für Java über eine Eingabeaufforderung mit der Option -Icon zu starten, und überprüfen Sie die Konsole auf Fehlernachrichten. Wenn der Arbeitsplatz (ide.icx) beschädigt ist, müssen Sie ihn möglicherweise ersetzen. Sie können auch die Datei ide.ini so editieren, dass sie auf ein lokales Repository verweist. Entfernen Sie dazu den Verweis auf eine **Server-Adresse** und ändern Sie den Wert im Feld **DefaultName** (Standardname) so, dass er auf ein lokales Repository verweist. Nach dem Starten von VisualAge für Java können Sie manuell eine Verbindung zum gemeinsam benutzten Repository herstellen.
Beim Starten von VisualAge für Java erhalten Sie eine ähnliche Fehlernachricht wie folgende: “Fehler 58, Ungültige Verlagerung”.	Dieser Fehler tritt auf, wenn das VisualAge für Java-Image nicht in den Speicher geladen werden kann. Versuchen Sie, die Datei ide.icx zu ersetzen. Funktioniert dies nicht, müssen Sie VisualAge für Java möglicherweise erneut installieren.
VisualAge für Java blockiert, wenn Sie versuchen, das Programm zu starten.	Möglicherweise ist die Datei ide.icx beschädigt. Ersetzen Sie die Datei durch eine Sicherungskopie.

3.2 Repositoryprobleme und -fehler behandeln

In der folgenden Tabelle werden einige Repositoryprobleme und -fehler aufgeführt, die auftreten können. Die IDE-Onlinehilfefunktion oder die IDE PDF im Verzeichnis ’pdf’ auf der Produkt-CD enthalten weitere Informationen zur Ausführung der in der Spalte ’Lösung’ unten aufgeführten Tasks.

Problem	**Lösung**
Mit dem Repository treten verschiedene Fehler auf wie beispielsweise folgende: • Code kann nicht versioniert werden • Wenn Sie versuchen, Code aus dem Repository in den Arbeitsbereich zu laden, stürzt der Arbeitsbereich ab. • Beim Komprimieren des Repositorys geht versionierter oder freigegebener Code verloren.	Möglicherweise ist Ihr Repository beschädigt. Installieren Sie ein neues Repository und versuchen Sie, den Code aus dem beschädigten Repository in das neue Repository zu importieren.
Wenn Sie VisualAge für Java herunterfahren und erneut starten, stellt das Programm eine Verbindung zum lokalen Repository her, obwohl es vor dem Herunterfahren mit einem gemeinsam benutzten Repository verbunden war.	Die Ursache hierfür liegt möglicherweise in einem DNS-Problem. Überprüfen Sie die Datei ide.ini, um zu ermitteln, ob ein Einheitenname auf die Server-Maschine verweist. Falls ja, ändern Sie den Namen in die IP-Adresse.

4.0 Dokumentation

4.1 Aktualisierungen der englischen Dokumentation

4.1.1 Task-Datei - Code implementieren

Die folgenden Laufzeitdateien sind nicht in VisualAge für Java, Version 4.0 enthalten:

- eab/runtime20/ivj2cpp.zip (C ++ Access Builder)
- ivjpb35.jar * (Persistence Builder)
- ivjpb35wsa35.jar * (Persistence Builder)

* Sie verfügen nur über diese Dateien, wenn Sie eine Migration von VisualAge für Java, Version 3.5. durchgeführt haben

Die folgenden Laufzeitdateien sind zwar in VisualAge für Java, Version 4.0 enthalten, sie sind jedoch nicht in der Datei ″Code implementieren″ (Deploying code) aufgelistet:

SAP Connector für Enterprise Access Builder

- eab/HP-UX/libivjsij35.sl,
- eab/HP-UX/libivjsid35.sl
- eab/HP-UX/librfccm.sl

Der IMS TOC Connector heißt nun IMS-Konnektor und die entsprechende Laufzeitdatei heißt imsconn anstatt imstoc.

Dieses Release von VisualAge für Java beinhaltet eine Beta-Version von mehreren Komponenten (einschließlich Konnektoren) der Java 2 Platform, Enterprise Edition (J2EE (TM)). Sun Microsystems Inc. hat dieses J2EE-Komponenten noch nicht offiziell freigegeben. Die Laufzeitdateien für die J2EE-Konnektoren befinden sich in eab\runtime35 und IBM Connectors\classes *nachdem* Sie die Beta-Komponenten installiert haben. Die Dateien ctgclient.jar und ctgserver.jar können nur mit dem CCF CICS Connector verwendet werden. Sie können nicht mit dem J2EE-CICS Connector verwendet werden, da sie die J2EE-Funktion nicht unterstützen. Darüber hinaus werden nicht alle J2EE-Laufzeitdateien unter Windows 2000 unterstützt.

Weitere Informationen zu den J2EE-Komponenten enthalten die Online-Hilfefunktion für Die Enterprise Access Beans und die Release-Hinweise.

4.2 Unterbrochene Verknüpfungen im Navigationsteilfenster

In der Professional Edition von VisualAge für Java für alle übersetzten Sprachen funktionieren im Navigationsteilfenster die folgenden Verknüpfungen nicht:

- ...für Ressourcendateien (dies ist eine Verknüpfung in der IDE Konzept-Hilfe)
- Scratch-Editionen (eine Verknüpfung in der IDE Konzept-Hilfe)
- Einen Standardarbeitsplatz bereitstellen (eine Verknüpfung in der IDE Aufgaben-Hilfe)

Diese Verknüpfungen mit Informationen zur Entwicklung im Team funktionieren nicht, da die Team-Komponente nicht in der Professional Edition von VisualAge für Java enthalten ist. Keine der in diesen Dateien enthaltenen Informationen ist für die Entwicklung in der Professional Edition von VisualAge für Java von Belang.

4.3 Nicht übersetzte Änderungen an der Dokumentation

Es folgt eine Liste der kritischen Änderungen an den Onlineinformationen, die nicht übersetzt worden sind.

1. Die folgenden Konzeptdateien wurden geändert:
   - ″Den richtigen Debugger für das Programm auswählen″ (Choosing the right debugger for the program)
   - ″Editionen und Versionieren″ (Editions and versioning)
   - ″Schrittweise Kompilierung″ (Incremental compilation)
   - ″Projekte und andere Programmelemente″ (Projects and other program elements)
   - ″Repository″ (Repository)
   - ″Ressourcendateien und -verzeichnisse″ (Resource files and directories)
   - ″Neuerungen in der VisualAge für Java-IDE″ (What's new in the VisualAge for Java IDE)
   - ″Ungelöste Probleme″ (″Unresolved problems″)
   - ″SmartGuide 'Servelet erstellen'″ (″Create Servlet SmartGuide)
   - ″Server-Aspekte in der Team-Entwicklung″ (″Server considerations in team development″)
2. Die folgenden Referenzdateien wurden geändert:
   - ″Repository-Dateien″ (Repository files)
   - ″Tasten für Direktaufruf″ (Shortcut keys)
   - ″Symbole für die VisualAge für Java-IDE″ (VisualAge for Java IDE symbols)
   - ″Code-Assistent″ (″Code assist″)
3. Die folgenden Task-Dateien wurden geändert:
   - ″Projekte und Pakete aus dem Repository hinzufügen″ (Adding projects and packages from the repository)
   - ″Code anwenden″ (Deploying code)
   - ″Unterbrechungspunkte im Quellencode setzen″ (Setting breakpoints in source code)
   - ″Methoden zum Stepping-Through″ (Stepping through methods)
   - ″Ein Programmelement versionieren″ (Versioning a program element)
   - ″Code exportieren″ (Exporting code)
   - ″IBM-Service-Korrekturen mit dem Fix Manager durchführen″ (Applying IBM service fixes with Fix Manager)
   - ″Probleme finden und korrigieren″ (Finding and fixing problems)
   - ″Text in einem Quellen-Teilfenster suchen ″ (Searching for text in a source pane)
   - ″Programmelemente drucken″ (Printing program elements)
   - ″Ressourcendateien exportieren″ (Exporting resource files)
   - ″In ein anderes Repository exportieren″ (Exporting to another repository)
   - ″Den Arbeitsbereich durchsuchen″ (Browsing the workspace)
   - ″Den Cache einrichten und löschen″ (Setting up and clearing the cache)
   - ″Unterbrechungspunkte in externen Klassen (IDE-Debugger) setzen″ (Setting breakpoints in external classes (IDE debugger))
   - ″Unterbrechungspunkt im Quellcode (IDE-Debugger) setzen″ (Setting breakpoints in source code (IDE debugger))

- ″Code exportieren″ (Exporting code)
- ″HTML-Dokumentationen für Klassen generieren″ (Generating HTML documentations for classes)
- ″Programmelemente drucken″ (Printing program elements)
- ″Mit Code-Fragmenten experimentieren″ (Experimenting with code fragments)
- ″Servlets mit dem SmartGuide 'Servlet erstellen' erstellen″ (Creating servlets using the Create Servlet SmartGuide)

4. Die folgenden Task-Dateien wurden neu hinzugefügt.
   - ″Eine Lösung finden″ (Creating a solution)
   - ″Code im Quellenansichts-Fenster editieren″ (Editing code in the Source View window)
   - ″Eine serielle UID generieren″ (Generating a serial UID)
   - ″Die Konfigurationsdatei des SmartGuides 'Servlet erstellen' ändern″ (Modifying the Create Servlet SmartGuide configuration file)
5. Die folgenden Referenzdateien wurden neu hinzugefügt.
   - ″Java-Compiler″ (Java compiler)
6. Die folgenden Konzeptdateien wurden neu hinzugefügt.
   - ″Platten- und Speicher-Cache″ (Disk and memory cache)
   - ″Java-Klassenbibliotheken″ (Java class libraries)
   - ″Klassenladeprogramm″ (Class loader)
   - ″Quellenansichts-Fenster″ (Source view window)
7. Die folgenden Task-Dateien wurden entfernt (nicht länger gültig):
   - ″Ressourcendateien gemeinsam benutzen″ (Sharing resource files)
8. Die Option ″Ressourcen und Pfade″ (″Resources and Paths″) ist jetzt ″Ressourcen″ (″Resources″).
9. Die folgenden F1-Hilfedateien wurden geändert:
   - SmartGuide ″Exportieren″ - Verzeichnisoptionen (Export SmartGuide - Directory Options)
   - SmartGuide ″Exportieren″ - Jar-Dateioptionen (Export SmartGuide - Jar file options)
   - ″Importieren aus einem Verzeichnis″ (Import from a directory)

   Die Kontexthilfedateien sind nicht in der englischen PDF-Datei enthalten.

VisualAge(R) für Java(TM)

Version 4.0

Visual Composition Editor

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einschränkungen und bekannte Probleme
1.1 Morphen von Beans
1.2 1.2 Visuelle Elemente mit Array-Eigenschaften
1.3 Klassenpfadinformationen gingen beim Export einer Bean verloren
1.4 JAR/zip-Dateien, die zum Klassenpfad des hinzugefügt werden, werden

nicht sofort angewendet
1.5 Verbinden von Ereignissen zum Warten oder zum Benachrichtigen von Methoden
2.0 Hinweise und Tipps
2.1 main() bei kopierten/verschobenen/umbenannten Klassen löschen
2.2 Applets verwenden und manuipulieren

1.0 Einschränkungen und bekannte Probleme

1.1 Morphen von Beans

Beim Morphen von Beans wird die Reihenfolge der Verbindungen von der Bean möglicherweise nicht eingehalten. Wählen Sie 'Verbindungen umordnen von' im Kontextmenü der gemorphten Bean aus und ordnen Sie die Verbindungen um.

1.2 Visuelle Elemente aus Quellen mit Array-Eigenschaften erstelle

Beim Wiederherstellen der visuellen Elemente von Beans, die Array-Typ-Eigenschaften besitzen, müssen Sie sicherstellen, dass der Leaf-Typ über einen registrierten Eigenschaftseditor verfügt. Andernfalls kann VisualAge nicht den richtigen Initialisierungscode für diese Eigenschaft generieren. Sie müssen eigene Codedatenströme in den Array-Editor eingeben.

1.3 Klassenpfadinformationen gingen beim Export einer Bean in ein Verzeichnis oder ein JAR verloren

Wenn Sie eine Bean in ein Verzeichnis oder eine JAR-Datei exportieren und sie anschließend importieren, geht die mit der Bean gespeicherte Klassenpfadinformation verloren. Beim Ausführen der importierten Bean kann es sein, dass Sie eine NoClassDefFound-Ausnahmebedingung erhalten. Vor Ausführen der Klasse müssen Sie den Klassenpfad definieren.

1.4 JAR/zip-Dateien, die zum Klassenpfad des Arbeitsbereichs hinzugefügt werden, werden nicht sofort angewendet

Wenn Sie über das Dialogfenster **Fenster -> Optionen...** JAR- oder Zip-Dateien zum Klassenpfad des Arbeitsbereichs hinzufügen, werden diese nicht unmittelbar angewendet. Fügen Sie beliebige Projekte oder Pakete hinzu oder löschen Sie sie und führen Sie einen Neustart der IDE durch, damit die Änderungen in Kraft treten.

1.5 Verbinden von Ereignissen zum Warten oder zum Benachrichtigen von Methoden

Wenn Sie im Visual Composition Editor erneut ein Ereignis mit einer der folgenden Methoden verbinden...

- notify()
- notifyAll()
- wait()
- wait(long)
- wait(long, int)

...müssen Sie 'synchronize' der tatsächlichen Ereignis-zu-Code-Methode hinzufügen, um eine java.lang.IllegalMonitorStateException zu vermeiden.

Wenn die Signatur Ihrer Verbindung beispielsweise
`private void connEtoM1(java.awt.event.ActionEvent arg1)`

lautet, ändern Sie die Anweisung wie folgt:
`private synchronized void connEtoM2(java.awt.event.ActionEvent arg1)`

2.0 Hinweise und Tipps

2.1 2.1 main() bei kopierten/verschobenen/umbenannten Klassen löschen

Wenn Sie eine visuell erstellte Klasse kopiert, verschoben oder umbenannt haben, löschen Sie main() und sichern Sie die Klasse. Durch Löschen von main() entfernen Sie veralteten Code, der andernfalls nicht überschrieben wird. Wenn die veraltete main() gelöscht ist, erstellt VisualAge für Java erneut eine gültige main(), wenn Sie die Klasse sichern.

2.2 Applets verwenden und manuipulieren

Um eine JMenuBar zu einem JApplet hinzuzufügen, müssen Sie zunächst ihr contentPane zum freien Entwurfsbereich ziehen. Hierdurch können Sie die JMenuBar an das JApplet übergeben, und nicht an die JPanel, die das contentPane ist.

Wenn Sie ein JApplet als Unterklasse erstellt haben und seine Größe oder Position auf der Seite 'Visuelle Erstellung' ändern möchten, müssen Sie sicherstellen, dass Sie das JApplet und nicht sein contentPane auswählen und bearbeiten. Da das contentPane des JApplet das JApplet vollständig verdeckt, müssen Sie das JApplet mit Hilfe der Beans-Liste auswählen und anschließend die Auswahlleisten und Rahmen manipulieren.

Diese Anwendung der Bean-Liste ist in vielen Situationen nützlich, in denen eines oder mehrere der Kinder eines Containers denselben vollständig verdecken.

VisualAge(R) für Java(TM)

Version 4.0

Java-Compiler und zugehörige Abschnitte

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Compiler - Einschränkungen und bekannte Probleme
    1.1 Debug-Attribute
    1.2 Standardpakete dürfen keine Klassen mit demselben Namen enthalten

1.0 Compiler - Einschränkungen und bekannte Probleme

1.1. Debug-Attribute

Der VisualAge für Java—Compiler generiert Debug-Attribute (falls angefordert).

Ein Java-Compiler muss verhindern, dass ein Benutzercode explizit eine nicht initialisierte lokale Variable liest. Ein Programm kann jedoch in einem Status unterbrochen werden, in dem einige Variablen zwar noch nicht gesetzt jedoch schon deklariert sind (Benutzerunterbrechung, ...). Ein Debugger versucht möglicherweise, alle sichtbaren Variablen an diesem Punkt anzuzeigen, indem er die zu diesem Zweck in .class-Dateien aufgezeichneten Debug-Attribute verwendet. Diese Debug-Attribute enthalten den Sichtbarkeitsbereich für jede lokale Variable. VisualAge für Java geht davon aus, dass jede Variable von ihrer Deklaration bis zum Ende ihres Definitionsbereichs sichtbar ist, während andere Compiler sie möglicherweise nur einmal als sichtbar anerkennen, wenn sie initialisiert wird.

Aus diesem Grund schlagen einige externe Debugger bei der VisualAge für Java-Ausgabe möglicherweise fehl bei dem Versuch, Variablen anzuzeigen, die noch nicht an einen Wert gebunden sind (z. B. bei dem Versuch, den Inhalt eines Array anzuzeigen, der noch nicht definiert ist). Der VisualAge für Java-Debugger arbeitet in diesem Fall richtig.

Aufgrund des VisualAge für Java-Algorithmus zum Zuordnen von Sichtbarkeitsbereichen kann ein Debugger lokale Variablen bei der Verwendung der VisualAge für Java-Ausgabe anders ordnen als bei der Verwendung der Ausgabe eines anderen Compilers.

1.2 Standardpakete dürfen keine Klassen mit demselben Namen enthalten

Befinden sich in einem Standardpaket in zwei oder mehr Projekten im Arbeitsbereich Klassen mit identischen Namen, wird möglicherweise eine java.lang.ArrayStoreException()-Ausnahmebedingung ausgegeben, wenn Sie eine der betreffenden Klassen ausführen. Dieses Problem können Sie umgehen, indem Sie die Klasse, die Sie ausführen möchten, erneut kompilieren. Hierzu müssen Sie die Klasse aus dem Arbeitsbereich löschen und sie anschließend erneut hinzufügen.

VisualAge(R) für Java(TM)

Version 4.0

Java-Klassenbibliothek und Programmausführung

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einführung
2.0 Die Klassenbibliotheken (Class Libraries)
    2.1 Referenzobjekte
    2.2 Paket java.awt
        2.2.1 Grafiken
    2.3 Paket sun.tools.jar
    2.4 Das Ändern von Java-Klassenbibliotheken
    2.5 Applet-Sicherheit
    2.6 RMI
3.0 Programmausführung
    3.1 Java-Threads
        3.1.1 Threads und JNI
        3.1.2 Threads im Debugger beenden
    3.2 Mehrere virtuelle Maschinen simulieren
    3.3 Java-Programme ausführen und einen Typ ändern

3.4 Objektbeendigung
3.5 Speicherbegrenzung
3.6 JNI
3.6.1 Nicht unterstützte JNI-API

1.0 Einführung

Diese Release-Hinweise enthalten Informationen über die Klassenbibliotheken, die mit VisualAge für Java, Version 4.0 geliefert werden. Darüber hinaus beschreiben diese Hinweise, wie Programme in der VisualAge-Umgebung ausgeführt werden.

2.0 Die Klassenbibliotheken - Einschränkungen und bekannte Probleme

2.1 Referenzobjekte

Derzeit ist nur java.lang.ref.WeakReference voll implementiert. Andere Referenzobjekte werden vom Abfallsammler als starke Referenzen behandelt.

2.2 Paket java.awt

2.2.1 Grafiken

Nachfolgend sind die bekannten Einschränkungen und Auslassungen in der AWT-Grafikunterstützung beschrieben, die von den Klassenbibliotheken von VisualAge Version 4.0 bereitgestellt wird. Einige dieser Punkte beschreiben Plattformverhalten, über das Anwendungsprogrammierer informiert sein müssen.

- **Interner Fehler beim Ausführen von Code zum Laden von Bildern**
  Dieser Fehler kann vorkommen, wenn Sie mehrere Programme ausführen, mit denen Bilder geladen werden. Code, der Bilder abruft, muss vom nativen Code wieder Java aufrufen. Dies kann gelegentlich fehlschlagen, wenn das System unter extrem hoher Belastung steht.
- **Allgemeiner Zugriffsfehler beim Ausführen vieler Kopien von demos.Java2Demo**
  Die Ausführung von VisualAge für Java kann nach dem Ausführen mehrerer Exemplare von
  demos.Java2Demo aufgrund einer Einschränkung im Low-Level-Font-Code fehlschlagen.

2.3 Paket sun.tools.jar

Durch Ausführen des jar-Dienstprogramms von der IDE werden Platzhalterzeichen-Argumente nicht wie nachfolgend umgesetzt:

```
jar cvf test.jar *.class
```

Dies ist korrektes Verhalten — in der Sun-Implementierung stellt die Datei JAR.EXE die Verwendung von Platzhalterzeichen bereit, nicht der Java-Code.

2.4 Das Ändern von Java-Klassenbibliotheken

In VisualAge für Java, Version 4.0 können bestimmte Pakete und Klassenbibliotheken ersetzt oder gelöscht werden. Ebenso können Sie Ihre eigenen Pakete oder Klassen einem Projekt hinzufügen.

Während einige der Pakete und Klassen verändert oder verschoben werden können, sollten sie unter keinen Umständen in ein anderes Projekt im Arbeitsbereich verschoben werden. Geschieht dies dennoch, so ist das Verhalten der Pakete und Klassen nicht definiert.

2.5 Applet-Sicherheit

Dieser Abschnitt beschreibt das Verhalten unter VisualAge-Versionen, die mit dem JDK-Datenstrom 1.1x von Sun kompatibel sind.

Applets werden innerhalb der Umgebung bei aktivem AppletSecurity-Manager ausgeführt. Dieser Sicherheitsmanager stellt standardmäßig nur sehr eingeschränkten Netzzugriff zur Verfügung. Dieser Zugriff kann über das Applet-Viewer-Eigenschaftenmenü geändert werden.

2.6 RMI

Der RMIRegistry-Klassenpfad ist erforderlich. Die RMI-Registry verwendet den Klassenpfad des Arbeitsbereichs, der das Projekt mit den Stub-/Skeleton-Klassen enthalten muss, da ansonsten die Ausnahmebedingung 'NoClassDefFoundError' auftritt. Eine weitere Auswirkung besteht darin, dass andere Programme, die nach der Änderung des Klassenpfads (classPath) des Arbeitsbereichs gestartet werden, diesen Eintrag ebenfalls im Klassenpfad haben.

3.0 Programmausführung

Achtung: Die in diesem Abschnitt beschriebenen Aspekte der Ausführungsumgebung können sich in zukünftigen Releases ändern. Aus diesem Grund sollten Sie keinen Code implementieren, der auf dem hier beschriebenen Verhalten beruht.

3.1 Java-Threads

Java-Threads sind nicht als Betriebssystem-Threads implementiert. Grundsätzlich wird die gesamte Umgebung als einzelner Betriebssystem-Thread ausgeführt, und Java-Threads werden von der Umgebung selbst in der Software implementiert. Threads werden auf der Basis von Zeitscheiben von ungefähr 10 Millisekunden geplant. Dieser Wert kann nicht geändert werden.

Obwohl die Prioritäten von Java-Threads von der in java.lang.Thread verfügbaren API geändert werden können, werden sie intern so geplant, als hätten sie alle dieselbe Priorität. Die IDE selbst wird ebenfalls als Thread ausgeführt (was für den Anwendungscode nicht sichtbar ist), aber bei einer höheren Priorität. Hierdurch werden andere Threads möglicherweise "eingefroren", wenn IDE-Operationen ausgeführt werden.

Auf diese Weise wird es der IDE-UI ermöglicht, auf Benutzeraktionen besser zu reagieren. Dies bedeutet jedoch auch, dass bei einem Blockieren der IDE-UI aus einem unbestimmten Grund die Umgebung so reagiert, als wäre sie blockiert.

Typische Aktionen, die eine solche Situation hervorrufen, sind das Warten auf Betriebssystemblockungsaufrufe oder die Blockung von Überwachungen (d. h. synchronisierten Methoden). Auf diese Weise kann Java-Code, der vom IDE-UI-Thread ausgeführt wird, möglicherweise die Umgebung sperren. Normalerweise tritt dies nicht ein, es kann jedoch beim Prüfen von synchronisierten Methoden vom Debugger und bei bestimmten, vom Visual Composition Editor ausgeführten Operationen eintreten.

3.1.1 Threads und JNI

Obwohl Threads in der IDE nicht in realen Betriebssystem-Threads implementiert sind, sperrt ein JNI-Aufruf, der in einem Betriebssystemblockungsaufruf enden kann, nicht das gesamte VM. Es wird ein proprietärer Mechanismus verwendet, so dass die anderen Java-Threads noch ausgeführt werden und das gleichzeitige Ausführen von realen Betriebssystem-Threads simulieren.

3.1.2 Threads im Debugger beenden

Beim Beenden von Threads im Debugger ist Vorsicht geboten, da dies zu gegenseitigem Sperren führen oder verhindern kann, dass eine Anwendung Ereignisse verarbeitet. Der standardmäßige AWT-Ereignis-Dispatch-Thread für ein Java-Programm (das über AWT-Fenster verfügt) wird in der Liste von verfügbaren Threads angezeigt und wenn er beendet wird, antwortet dieses bestimmte Programm nicht mehr auf UI-Ereignisse (Bewegungen/Eingaben/Anklicken mit der Maus, enter). Dieses Java-Programm muss dann mit Hilfe des Debuggers beendet werden, damit das/die Fenster geschlossen wird/werden.

Wenn Sie den Visual Composition Editor (VCE) verwenden und "Systemprogramme anzeigen" auf der Seite "Debuggen" im Dialog "Optionen" ausgewählt ist, können Sie den AWT-EventQueue-Thread für das von Ihnen visuell erstellte Programm sehen. Hier gelten dieselben Einschränkungen - stellen Sie sicher, dass Sie keinen der Threads oder das Programm selbst beenden. Stellen Sie sicher, dass Sie für dieses bestimmte Programm nur den VCE selbst verwenden. Wenn Sie dies nicht befolgen, können interne Fehler auftreten.

3.2 Mehrere virtuelle Maschinen simulieren

Das Scrapbook kann mehrere Seiten enthalten, von denen sich jede Seite wie ein vollkommen separates Java-Programm verhält, das auf einer eigenen virtuellen Maschine ausgeführt wird. Code, der auf einer Seite ausgeführt wird, hat keine Nebeneffekte auf Code, der auf anderen Seiten ausgeführt wird. Wenn Sie beispielsweise über Client/Server-Programme verfügen und sie in derselben Maschine testen möchten, müssen Sie daher keine zwei separaten Umgebungen starten. Sie müssen lediglich eine Seite im Scrapbook öffnen und Code eingeben, um den Server aufzurufen. Server und Client verhalten sich so, als ob sie von zwei separaten virtuellen Maschinen aufgerufen worden wären.

Es folgt ein einfaches Beispiel: Wenn Sie eine Klasse X mit einem statischen Slot y haben, die wie folgt mit Null initialisiert wird:

```
public static String y = null;
```

und Sie eine neue Seite des Scrapbook öffnen und folgendes ausführen:

```
X.y = "abc"; com.ibm.uvm.tools.DebugSupport.inspect (X.y);
```

erhalten Sie einen Inspektor, der folgendes anzeigt:

```
"abc"
```

Gehen Sie nun zu einer anderen neuen Seite des Scrapbook und führen Sie folgendes aus:

```
com.ibm.uvm.tools.DebugSupport.inspect (X.y);
```

In diesem Inspektor ist der Wert von y immer noch Null.

Beachten Sie jedoch, dass beim Hinzufügen, Ändern oder Löschen von Methoden oder Klassen diese Änderungen global für die Umgebung gelten. Sämtliche Änderungen im Code an Methoden oder an der Form der Klassen sind also sofort auf allen Scrapbook-Seiten sichtbar, auf denen diese Klassen verwendet werden (und in allen zugehörigen Inspektoren, Debuggern, etc.). In einigen Fällen wird die Scrapbook-Seite je nach der Art der Änderung zurückgesetzt. Weitere Informationen hierzu finden Sie im Abschnitt 'Java-Programme ausführen und einen Typ ändern'.

Code, der von den Menüpunkten 'Ausführen - In Applet Viewer...' und 'Ausführen - Main ausführen...' ausgeführt wird, verhält sich erwartungsgemäß ebenfalls wie ein vollkommen separates Java-Programm, das auf einer eigenen virtuellen Maschine ausgeführt wird.

3.3 Java-Programme ausführen und einen Typ ändern

Sie können Methoden während der Ausführung eines Java-Programms ändern. Diese Änderungen beeinflussen unmittelbar das Verhalten der Anwendung, so dass Sie mit Änderungen experimentieren können. Wenn Sie bei einem Typ jedoch erhebliche Änderungen vornehmen (Felder hinzufügen/entfernen, usw.), werden alle Programme, bei denen dieser Typ initialisiert wurde, beendet. Wenn es sich um Scrapbook-Seiten handelt, werden die Seiten zurückgesetzt.

Änderungen an statischen Feldern führen nicht zu Programmbeendigungen oder der Zurücksetzung von Scrapbook-Seiten.

Ein spezieller Fall dieses Verhaltens ist die Methode finalize(). Ein Objekt wird nur beendet, wenn eine Methode finalize() (eine andere als die in Objekt) definiert wurde, als das Objekt instantialisiert wurde. Durch das Hinzufügen eines Finalizers zu ausgeführtem Code können alle neuen Objekte beendet werden, wenn sie unbrauchbar werden. Alle bereits zuvor bestehenden Objekte werden jedoch nicht beendet. Wenn ein Finalizer gelöscht wird, wird beim Erfassen der Objekte die nächste Implementierung der Superklasse finalize() verwendet (möglicherweise die in Objekt).

3.4 Objektbeendigung

Objekte, auf die nicht mehr verwiesen wird, können beendet werden, wenn sie die Methode finalize() implementieren. Es ist jedoch wichtig zu beachten, dass Benutzer jederzeit Finalizer schreiben können, die potentiell in einer Endlosschleife laufen können.

Ein Finalizer kann beispielsweise implementiert werden, um folgendes auszuführen:

```
while (true) {};
```

Hierdurch wird der Finalizer-Thread in eine Endlosschleife gestellt, in der er keine anderen Objekte mehr beenden kann.

Ein weiterer offensichtlicher Fall ist das gegenseitige Sperren, bei dem ein Finalizer versucht, eine synchronisierte Methode aufzurufen und sich bereits ein anderer, zum Objekt zugehöriger Thread in der Überwachung befindet. In diesem Fall muss der Finalizer warten, bis die Überwachung freigegeben wird. Wenn die Überwachung aufgrund von Programmfehlern nie freigegeben wird, wird der Finalizer ebenfalls endlos blockiert.

Programme können auch viel Unbrauchbares generieren. In diesem Fall müssen die Finalizer häufiger ausgeführt werden. Das Ausführen von Finalizern ist sehr wichtig in einer integrierten Umgebung wie der IDE, da sich auf ihr mehrere aktive Programme befinden können (z. B. mehrere Applets oder mehrere Seiten im Scrapbook), die mehrere virtuelle Maschinen simulieren, wie in Abschnitt 3.2 beschrieben.

Werden keine Finalizer ausgeführt, kann dies Verluste von Betriebssystemressourcen bedeuten. Aus diesem Grund wird bei der IDE darauf geachtet, so oft wie möglich Beendigungen durchzuführen.

Um mit den oben beschriebenen Einschränkungen umgehen zu können, reserviert die IDE einen Finalizer-Thread mit einer internen Priorität, die höher ist als die Priorität, mit der die Java-Programme normalerweise ausgeführt werden (Siehe Abschnitt 'Java-Threads'). Dieser Thread ist eine gewisse Zeit inaktiv, wird dann aktiv und beendet einige Objekte, und wird danach wieder inaktiv. Wenn viel Unbrauchbares erstellt wird, ist er zwischen den inaktiven Phasen länger aktiv (zum Beenden). Der Thread hat auch ein Zeitlimit beim Ausführen der Beendigung. Wenn Objekte in einem bestimmten Intervall nicht beendet werden können, wird der Finalizer-Thread beendet und es wird ein neuer erstellt. Daraufhin wird der gesamte Beendigungsmechanismus erneut gestartet. Momentan beträgt das Zeitlimit für die Beendigung fünf Sekunden, und der Finalizer wird jede Sekunde inaktiviert/aktiviert. Diese Werte können nicht geändert werden.

3.5 Speicherbegrenzung

Dieses Release von VisualAge besitzt ein aggressives Speicherzuordnungsschema, das mehrmals versucht, den Freispeicher zu komprimieren, bevor es beim Betriebssystem mehr Speicher anfordert. Darüber hinaus ist das Limit für das Zuordnen von Speicherplatz vom Betriebssystem relativ hoch. Wenn die Maschine die Speicherzuordnung überlastet hat, veranlaßt der Komprimierungsalgorithmus die Auslagerung. Wenn der Benutzer die für das Produkt verfügbare Speicherkapazität manuell reduzieren möchte (d. h., wenn er veranlaßt, dass die Ausnahmebedingung OutOfMemoryError früher ausgegeben wird), müssen Sie die folgenden beiden Textzeilen an das Ende der ide.ini-Datei im Programmverzeichnis anhängen:

[VM Options]
maximumMemoryLimit=ZZZ

Hierbei ist ZZZ die maximale Anzahl von BYTE, die dem Produkt zugeordnet werden dürfen. (Beispiel: maximumMemoryLimit=128000000 für ein ungefähres Speicherlimit von 128 MB).

Darüber hinaus ist zu beachten, dass VM beim Behandeln von OutOfMemoryError ein anderes Verhalten als das Sun-J2SDK(TM) aufweist. Der Versuch, nach dem Ausgeben der Ausnahmebedingung noch viele weitere Objekte zuzuordnen, kann zur abrupten Beendigung der gesamten IDE führen. Wenn Sie OutOfMemoryError-Nachrichten erhalten, sollten Sie die IDE schließen und das maximumMemoryLimit erhöhen.

3.6 JNI

3.6.1 Nicht unterstützte JNI-API
Die JNI-API-Aufrufmethoden sind nicht implementiert.

Dies ist eine bekannte Einschränkung in VisualAge für Java, Version 4.0.

Folgende Implementierungen sind vorhanden (JNI 1.2 APIs):
FromReflectedMethod
FromReflectedField
ToReflectedMethod
ToReflectedField

Folgende Implementierungen sind vorhanden:
AttachCurrentThread
DetatchCurrentThread

Folgende Implementierungen sind als NOP vorhanden (sie sind funktionsfähig, aber wirkungslos):
EnsureLocalCapacity
PushLocalFrame
PopLocalFrame

Folgende Implementierungen sind vorhanden und führen zum Versagen:
JNI_CreateJavaVM
DestroyJavaVM
JNI_GetDefaultJavaVMInitArgs

Folgende Implementierungen sind vorhanden, jedoch unvollständig:
JNI_GetCreatedJavaVMs - gibt entweder nichts oder die aktive VM zurück

VisualAge(R) für Java(TM)

Version 4.0

Online-Hilfe

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einführung
2.0 Einschränkungen und bekannte Probleme
    2.1 Unerwartete Ergebnisse bei der Satzsuche
3.0 Hinweise und Tipps
    3.1 Fehlerbehebung bei Problemen mit der Onlinehilfefunktion und NetQuestion
    3.2 Installation auf einem LAN-Laufwerk für Windows 98
    3.3 Installation und Konfiguration von TCP/IP
    3.4 Wechsel zu einer anderen Anschlussnummer
    3.5 Wechsel zu einem anderen Browser
    3.6 Manuelle Konfiguration von Proxy-Einstellungen
    3.7 Web-Browser unter Windows aufrufen

1.0 Einführung

Diese Release-Hinweise enthalten die Einschränkungen und Hinweise im Hinblick auf die Funktion der VisualAge-Hilfe und der NetQuestion-Such-Engine.

2.0 Einschränkungen und bekannte Probleme

2.1 Unerwartete Ergebnisse bei der Satzsuche

Der Parser des Suchindexprozesses von NetQuestion fügt Abschnitts- bzw. Kopfzeilenwechsel nicht an der korrekte Position ein. Anstatt einen Wechsel am Ende eines Abschnitts oder einer Kopfzeile einzufügen, wird der Wechsel nach dem ersten Wort des folgenden Abschnitts bzw. der folgenden Kopfzeile eingefügt. Dieser Fehler führt unter anderem dazu, dass die Such-Engine bei der Suche nach einem Satz, der das erste Wort eines Abschnitts oder einer Kopfzeile sowie darauffolgende Wörter enthält, nicht die Dokumente zurückgibt, die tatsächlich das Ziel der Suche enthalten. Eine weitere Folge dieses Fehlers besteht darin, dass die Such-Engine bei der Suche nach einem Satz, der das letzte Wort eines Abschnitts oder einer Kopfzeile gefolgt vom ersten Wort des folgenden Abschnitts oder der folgenden Kopfzeile enthält, Dokumente im Suchergebnis zurückgibt, die das Suchziel gar nicht enthalten.
Nehmen Sie als Beispiel ein Online-Hilfedokument mit dem HTML-Inhalt aus (1).

(1)

```
<p>eins zwei drei</p>
<p>vier fünf sechs</p>
```

Da die Satzwechsel nach “eins” und “vier” nicht korrekt positioniert sind, schlagen die folgenden gültigen Suchoperationen fehl: “eins zwei drei”, “eins zwei”, “vier fünf”. Bei den folgenden ungültigen Suchoperationen hingegen wird das Dokument aus (1) zurückgegeben: “drei vier”, “zwei drei vier”.

3.0 Hinweise und Tipps

3.1 Fehlerbehebung bei Problemen mit der Onlinehilfefunktion und NetQuestion

Falls Sie Ihr Problem nicht mit den Hinweisen und Tipps in diesem Dokument lösen können, lesen Sie bitte das in dieser Installation enthaltene Handbuch zur Fehlerbehebung der VisualAge-Hilfefunktion.

3.2 Installation auf einem LAN-Laufwerk für Windows 98

Wenn Sie das Produkt auf einem LAN-Laufwerk ″X:″ installieren und zuvor nicht NetQuestion installiert haben, schlägt der AUTOEXEC.BAT-Aufruf an das Suchsystem ″X:\IMNNQ_98\IMNENV.BAT″ fehl. Um diesen Fehler zu beheben, kopieren Sie “X:\IMNNQ_98\IMNENV.BAT” nach C: und editieren Sie die Datei AUTOEXEC.BAT so, dass “C:\IMNENV.BAT” aufgerufen wird.

3.3 Installation und Konfiguration von TCP/IP

TCP/IP muss installiert und konfiguriert sein, damit die Online-Hilfe richtig funktionieren kann. Unter Windows 98 muss TCP/IP wie folgt aktiviert sein:

1. Bei einer LAN-Adapter-Konfiguration:
   - DNS muss mit einem gültigen Host- und Domänennamen aktiviert sein.
2. Bei einer Wahl-Adapter-Konfiguration:
   - DNS muss inaktiviert sein.
   - Ihre TCP/IP-Adresse muss automatisch abgerufen werden.

Hinweis: Diese Konfigurationsoptionen gelten für alle TCP/IP-Adapter, auch dann, wenn die Optionen nur für diesen einen Adapter geändert wurden. Sie können nicht sowohl eine LAN-Konfiguration als auch eine Wahlkonfiguration verwenden, ohne zuvor eine Neukonfigurierung durchgeführt zu haben.

Die TCP/IP-Wahlnetzwerkeigenschaften für Ihre(n) Internet-Service-Provider (ISP) müssen so konfiguriert sein, wie dies durch den jeweiligen ISP dokumentiert ist. Die TCP/IP-Wahlnetzwerkeigenschaften setzen die TCP/IP-Eigenschaften außer Kraft, die für den Wahl-Adapter über das Symbol ’Netzwerk’ in der Systemsteuerung von Windows 98 konfiguriert wurden. Dieses außer Kraft Setzen tritt jedoch nur dann auf, wenn die TCP/IP-Eigenschaften für den Wahl-Adapter wie oben beschrieben konfiguriert sind.
Sie dürfen den DNS in den TCP/IP-Eigenschaften des Wahl-Adapters nicht aktivieren; weiterhin dürfen Sie nicht eine IP-Adresse in den TCP/IP-Eigenschaften des Wahl-Adapters angeben. Andernfalls tritt ein Konflikt mit der Konfiguration des Wahlnetzwerks für den ISP auf.

Die oben genannten Konfigurationen gelten nur für Windows NT 4. Wenn Sie VisualAge für Java auf einer eigenständigen Maschine ausführen, können Sie auch den MS Loopback Adapter ohne die beiden anderen Adapter ausführen.

3.4 Wechsel zu einer anderen Anschlussnummer

Die Hilfefunktion ist dem Anschluss 49213 zugeordnet, einer Nummer außerhalb der öffentlichen Anschlüsse, die TCP/IP zugeordnet sind. Wenn Sie mit einem anderen Produkt arbeiten, das diese Anschlussnummer verwendet, oder wenn Sie keine Verbindung zu diesem Anschluss herstellen können, können Sie den Anschluss für die Hilfefunktion wie folgt ändern:

1. Editieren Sie die Datei httpd.cnf, die sich im Verzeichnis IMNNQ_NT oder IMNNQ_95 befindet, und geben Sie eine verfügbare Anschlussnummer ein, vorzugsweise über 49000.
2. Bearbeiten Sie in der Datei c:\IBMJAVA\doc\ivjhlp.cfg (c:\IBMJAVA ist hierbei das Installationsverzeichnis von VisualAge für Java) den Wert HTML-_HOSTNAME. Dieser Wert lautet standardmäßig: localhost:49213. Ändern Sie diesen Wert, so dass die in Schritt 1 ausgewählte Anschlussnummer den Wert 49213 ersetzt.
3. Wechseln Sie in das Installationsverzeichnis von VisualAge für Java. Im Unterverzeichnis doc finden Sie die folgenden Dateien: hgssrch.htm und hgcsrch.htm hgssrch.htm, hgcsrch.htm, hgrban.htm
4. Editieren Sie die Datei hgssrch.htm mit einem Texteditor, und ändern Sie die folgende Zeile in der Datei, indem Sie die Anschlussnummer 49213 durch die in Schritt 1 ausgewählte Anschlussnummer ersetzen: <form action="http://localhost:49213/cgi-bin/ivjsrch.exe" METHOD="POST">
5. Wiederholen Sie diese Aktion für die Dateien hgcsrch.htm und hgrban.htm.
6. Aktualisieren Sie mit Hilfe Ihres Zip-Tools (Packerprogramm) die bearbeiteten Dateien in der Datei main.zip. **HINWEIS:** Sie müssen die neuen Dateien ohne Komprimierung und ohne Pfadinformationen aktualisieren/hinzufügen. Die Dateien in der Zip-Datei sind nicht komprimiert. Wenn Sie die Datei beim Aktualisieren komprimieren, sehen Sie beim erneuten Laden des Suchformulars nur Zufallszeichen.
7. Löschen Sie den Browser-Speicher und den Platten-Cache, und beenden Sie VisualAge für Java. Die Hilfe- und die Suchfunktion können nun mit dem neuen Anschluss verbunden werden. Funktioniert dies nicht, suchen Sie in der Prozessliste nach einem Prozess namens httpdl.exe. Beenden Sie diesen Prozess, verlassen Sie VisualAge für Java und führen Sie anschließend einen Neustart für VisualAge für Java aus.

3.5 Wechsel zu einem anderen Browser

Standardmäßig verwendet VisualAge für Java den Standard-Browser des Systems (Netscape oder Internet Explorer), um die Hilfe anzuzeigen. Wenn Sie einen anderen Browser als den Standard-Browser verwenden möchten, müssen Sie die Konfigurationsdatei der Hilfefunktion ändern. Das Ändern des Browsers von der Seite 'Hilfe' des Dialogs 'Optionen' aus funktioniert momentan nicht. Sie können den von der Hilfefunktion verwendeten Browser mit den folgenden Schritten ändern:

1. Bearbeiten Sie mit einem Texteditor die Datei C:\IBMVJava\doc\ivjhlp.cfg, wobei C:\IBMVJava das Installationsverzeichnis von VisualAge für Java ist.
2. Entfernen Sie das Zeichen # vom Anfang der Zeile #HTML_BROWSER="H:\Programs\Netscape\Communicator\Netscape.exe"
3. Ersetzen Sie H:\Programs\Netscape\Communicator\Netscape.exe durch den Pfad- und Dateinamen Ihres bevorzugten Browsers.
4. Sichern und schließen Sie die Datei ivjhlp.cfg.
5. Starten Sie VisualAge für Java erneut. Die Hilfefunktion sollte nun Ihren bevorzugten Browser verwenden.

3.6 Manuelle Konfiguration von Proxy-Einstellungen

Wenn Internet Explorer beim Starten der Online-Hilfe immer versucht, eine Verbindung zu Ihrem ISP herzustellen, müssen Sie diese Option im Internet Explorer inaktivieren. Je nachdem, welche Explorer-Version Sie verwenden, können Sie über das Menü "Tools", "Datei" oder "Ansicht" auf Ihre Internet-Optionen zugreifen.

3.7 Web-Browser über Windows aufrufen

Windows enthält zwei verschiedene Typen von Dateizuordnungen für web-basierte Dokumentation. Der erste Typ von Zuordnung ist für .htm- und .html-Dateien, die lokal aufgerufen werden (z. B. c:\index.htm). Der zweite Typ ist für nicht lokale Dateien im URL-Format (z. B. http://www.software.ibm.com/ad/vajava) und heißt "URL - Hypertext Transfer Protocol". VisualAge für Java verwendet für die meisten Hilfemenüpunkte die erste Zuordnung und die zweite Zuordnung für den Menüpunkt **Hilfe > Unterstützung**. Wenn ein Web-Browser installiert wird, werden normalerweise beide Zuordnungen automatisch konfiguriert.

Wenn eine Hilfeoption des Hilfemenüs von VisualAge für Java einen Web-Browser nicht öffnen kann, können Zuordnungen nicht richtig konfiguriert sein. Um festzustellen, ob die Zuordnungen in Windows richtig konfiguriert sind, gehen Sie zum Menü **Start** von Windows und wählen Sie **Ausführen** aus. Geben Sie den Namen einer .htm-Datei oder einer URL ein (z. B. 'http://www.software.ibm.com/ad/vajava'). In beiden Fällen sollte daraufhin ein Web-Browser geöffnet werden. Wenn das Öffnen des Web-Browsers fehlschlägt, sind die Zuordnungen nicht richtig konfiguriert.

Wenn kein Web-Browser geöffnet wird, können Sie entweder den Web-Browser erneut installieren, die Windows-Zuordnungen manuell konfigurieren oder in VisualAge für Java zur Seite **Hilfe** des Dialogs **Optionen** gehen (die Sie von der Workbench durch Auswählen von **Fenster -> Optionen** erreichen) und dort **Browser-Pfad angeben** anstatt **Dateizuordnung verwenden** auswählen.

Falls ein Web-Browser geöffnet wird, wenn Sie eine URL über das Windows-Menü **Start** eingeben, aber nicht, wenn Sie aus der Menüleiste der VAJava Workbench die Option **Hilfe** auswählen, lesen Sie bitte das in dieser Installation enthaltene Handbuch zur Fehlerbehebung der VisualAge-Hilfefunktion.

VisualAge(R) für Java(TM)

Version 4.0

Data Access Beans

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Voraussetzungen
2.0 Einschränkungen und bekannte Probleme
  2.1 Die Data Access Beans-Funktion aus dem Arbeitsbereich entfernen
  2.2 Keine automatische Scrolling-Unterstützung für umfangreiche Ergebnissätze
3.0 Hinweise und Tipps
  3.1 Die Data Access Beans-Funktion muss hinzugefügt werden, bevor die Beans verwendet werden können
  3.2 Den Datenbanktreiber in Ihren Klassenpfad aufnehmen

3.3 Arbeitscache vor dem Umschalten zwischen JDBC(TM)-Stufen löschen
3.4 Installation und Konfiguration von Merant SequeLink Server

1.0 Voraussetzungen

Um die Data Access Beans-Funktion anwenden zu können, müssen Sie über einen JDBC-fähigen Datenbanktreiber verfügen. Obwohl alle der in Version 3.5 neu hinzugefügten Funktionen von JDBC 2.0-Schnittstellen Gebrauch machen, muss ein JDBC 2.0-Treiber nur dann verwendet werden, wenn die oben erwähnte neue Funktion angewendet werden soll.

Wenn Sie DB2 als Ihre Datenbank verwenden und die neue Data Access Bean-Unterstützung für das Java DataSource-Objekt verwenden bzw. von der vollen Scrolling-Unterstützung für Ergebnisse bei limitierter Cache-Größe Gebrauch machen wollen, müssen Sie UDB, Version 7.1 oder eine höhere Version verwenden. DB2 verwendete JDBC 2.0-Treiber erstmals in Version 6. Diese Treiber enthielten allerdings noch keine Unterstützung für die neue Funktion.

2.0 Einschränkungen und bekannte Probleme

2.1 Das Entfernen der Data Access Beans-Funktion aus dem Arbeitsbereich verursacht einen Absturz der IDE.

Wenn Sie die Data Access Beans-Funktion dem Arbeitsbereich hinzufügen und später versuchen, diese Funktion wieder zu entfernen, stürzt die IDE ab. Wenn Sie den Arbeitsbereich nicht gesichert haben, verlieren Sie unter Umständen einige Daten. Es wird nicht empfohlen, die Funktion nach dem Hinzufügen wieder zu entfernen.

2.2 Keine automatische Scrolling-Unterstützung für umfangreiche Ergebnissätze

Die Online-Dokumentation für die Data Access Beans gibt an, dass Sie automatisch zwischen allen Zeilen umfangreicher Ergebnissätze hin- und herblättern können, selbst wenn Sie die Größe des Speicher-Cache durch Einstellen der Eigenschaft 'maximumPacketsInCache' Ihrer Select- oder ProcedureCall-Bean auf einen Wert ungleich 0 limitiert haben. Dies ist nicht korrekt.

Da die Mechanismen, die während der Bearbeitungszeit überprüfen, ob ein Datenbanktreiber der Stufe JDBC 2.0 entspricht, unter gewissen Bedingungen zu Blockierungen führten, könnte diese Unterstützung nicht automatisch eingestellt werden.

Wenn Sie volle Scrolling-Unterstützung für eine Select- oder ProcedureCall-Bean wünschen, deren Cache-Größe Sie limitiert haben, und Sie einen JDBC 2.0-Treiber verwenden, müssen Sie eine nicht dokumentierte Eigenschaft der Select- bzw. ProcedureCall-Bean mit der Bezeichnung 'setLargeResultSetScrollingEnabled' auf 'true' einstellen. Diese Eigenschaft ist nicht auf der Eigenschaftsseite für eine Select- bzw. ProcedureCall-Bean im Visual Composition Editor aufgeführt. Um die Eigenschaft auf 'true' einzustellen, müssen Sie eine Codezeile schreiben, die folgendem Beispiel entspricht:

```
ivjSelect1.setLargeResultSetScrollingEnabled(true);
```

Für einen Code, der unter Verwendung des Visual Composition Editors generiert wird, wird empfohlen, die Codezeile in den Benutzercode-Abschnitt der generierten Methode getSelect1() (oder getXXX(), wobei XXX den Namen Ihrer SelectBean darstellt) zu schreiben.

3.0 Hinweise und Tipps

3.1 Die Data Access Beans-Funktion muss hinzugefügt werden, bevor die Beans verwendet werden können

Wenn die Beans-Palette im Visual Composition Editor keine Bean-Kategorie namens “Datenbank” (“Database”) enthält, wurde die Data Access Beans-Funktion noch nicht hinzugefügt.

Um die Funktion hinzuzufügen, verwenden Sie das Menü **QuickStart** (F2) aus der Workbench. Wenn das Menü ″Quick Start″ angezeigt wird, heben Sie ″Funktionen″ hervor, wählen Sie ″Funktion hinzufügen″ aus, und klicken Sie auf ″OK″. Heben Sie die Data Access Beans aus der Funktionsliste hervor, und klicken Sie auf ″OK″.

3.2 Den Datenbanktreiber in Ihren Klassenpfad aufnehmen

Wenn Sie eine Anwendung ausführen, die die Data Access Beans verwendet, und Sie erhalten eine Nachricht, dass die angegebenen JDBC-Treiber nicht gefunden werden können, müssen Sie den Klassenpfad so abändern, dass er die JDBC-Treiberklassen enthält.

Sie können Ihren Treiber entweder dem Klassenpfad des Arbeitsbereichs hinzufügen (unter Verwendung der Kategorie ″Ressourcen″ im Menü ″Optionen...″) oder Sie können den Treiber dem Klassenpfad der jeweiligen Anwendungen hinzufügen, die den Treiber benötigen (unter Verwendung der Seite ″Klassenpfad″ im Notizbuch für Eigenschaften der Anwendung).

Wenn Sie beispielsweise die DB2 JDBC-Treiber dem Klassenpfad Ihres Arbeitsbereichs hinzufügen wollen, wählen Sie zunächst die Menüpunkte ″Fenster -> Optionen...″ im aktiven Fenster aus. Wählen Sie anschließend die Kategorie ″Ressourcen″ aus, und geben Sie den vollständigen Pfad für die Datei ’db2java.zip’ in das Feld des Klassenpfads im Arbeitsbereich ein:

C:\SQLLIB\java\db2java.zip;

(Ersetzen Sie hierbei C:\SQLLIB durch die Position Ihres SQLLIB-Verzeichnisses.)

3.3 Arbeitscache vor dem Umschalten zwischen JDBC(TM)-Stufen löschen

Wenn Sie VisualAge für Java verwendet haben, um auf einen Datenbanktreiber der Stufe JDBC 1.0 zuzugreifen, und anschließend auf einen Treiber der Stufe JDBC 2.0 überwechseln, der dieselben Klassennamen verwendet, müssen Sie den Platten-Cache und den Speicher-Cache von VisualAge für Java löschen, bevor Sie die neue Stufe verwenden können. (Sie müssen den Cache auch dann löschen, wenn Sie in die andere Richtung umschalten wollen, also von der Stufe JDBC 2.0 nach JDBC 1.0).

Um den Inhalt des Caches zu löschen, wählen Sie zunächst die Menüpunkte ″Fenster -> Optionen...″ im aktiven Fenster aus. Wählen Sie anschließend die Kategorien ″Allgemein -> Cache″ aus, und klicken Sie auf die Knöpfe ″Platten-Cache löschen″

und ″Speicher-Cache löschen″ und dann auf ″OK″. Sie müssen einen Systemabschluß für VisualAge für Java und anschließend einen Neustart durchführen, damit die Änderungen wirksam werden.

Wenn Sie den Cache vor der Verwendung der neuen Stufe des Datenbank-Treibers nicht löschen, kann Ihr Arbeitsbereich beschädigt werden und muss unter Umständen neu erstellt werden.

Da die DB2-Datenbanktreiber für JDBC 1.0 und JDBC 2.0 dieselben Klassennamen verwenden, gelten die vorstehenden Hinweise auch für Anwender von DB2.

3.4 Installation und Konfiguration von Merant SequeLink Server

Führen Sie die folgenden Schritte aus, um SequeLink Server von der CD für zusätzliche Funktionen (Additional Features) von VisualAge für Java zu installieren. Für diese Art von Installation müssen Sie keinen Registrierungsschlüssel eingeben. In einigen Schritten wird auf das Installationshandbuch von SequeLink verwiesen. Dieses Handbuch können Sie über dieSite für Merant-Dokumentation abrufen. HINWEIS: Detaillierte Informationen zur Position von Merant SequeLink Server finden Sie im Installations- und Migrationshandbuch.

Installation unter Windows NT oder Windows 2000

1. Legen Sie die CD für zusätzliche Funktionen (Additional Features) von VisualAge für Java in das CD-ROM-Laufwerk ein. (Falls nach dem Einlegen der CD automatisch das Installationsfenster angezeigt wird, klicken Sie 'Beenden' an, um das Fenster zu schließen. OEM SequeLink Server kann nicht mit dem Standardinstallationsprogramm installiert werden.)
2. Navigieren Sie zum Oracle- oder zum SQLServer-Server-Treiberordner. Navigieren Sie unter Windows NT beispielsweise zu einem der folgenden Ordner:
   - Navigieren Sie für den Oracle-Server-Treiber zu folgendem Ordner: Merant\SequeLinkServer\NT\Oracle
   - Navigieren Sie für den SQLServer-Server-Treiber zu folgendem Ordner: Merant\SequeLinkServer\NT\Mssql
3. Führen Sie das Konfigurationsprogramm mit dem folgenden Parameter aus: setup.exe /v"IPE=NO"
   Sie werden nicht aufgefordert, einen Registrierungsschlüssel einzugeben. Das Konfigurationsprogramm erstellt einen Standardservice SequeLink.
4. Nehmen Sie die Konfiguration vor, die zum Herstellen einer Verbindung zwischen SequeLink und Ihrer Datenbank erforderlich ist.

Merant SequeLink Server wird unter Windows 98 nicht unterstützt.

Konfiguration der WebSphere-Testumgebung für den Merant SequeLink-Treiber

VisualAge für Java 4.0 umfasst die Möglichkeit, einen Merant SequeLink-Treiber mit Oracle und Microsoft SQLServer zu verwenden.

Die folgenden Anweisungen beschreiben, wie die IBM WebSphere-Testumgebung für die Verwendung des Merant SequeLink-Treibers konfiguriert werden muss. Dabei wird davon ausgegangen, dass die folgenden Aktionen bereits ausgeführt wurden:

- Installieren von VisualAge für Java 4.0
- Installieren von Oracle oder Microsoft SQL Server und des entsprechenden Merant SequeLink-Server auf derselben Maschine

Darüber hinaus müssen Sie die folgenden Schritte ausführen, um den Merant-Treiber verwenden zu können:

1. Fügen Sie die folgenden JAR-Dateien zum Klassenpfad Ihres Arbeitsbereichs hinzu:
   project resources\IBM WebSphere Test Environment\lib\sljc.jar und sljcx.jar
2. Konfigurieren Sie die EJB-Server-Eigenschaften so, dass sie die folgenden URL-Informationen definieren:
   <datenbankpräfix>:jdbc:sequelink://<*hostname*>:<*anschluss*>
   Dabei ist ja nach Ihrer Umgebung <*hostname*> der Kurzname des Hosts (nicht der vollständig qualifizierte Host-Name) und <*anschluss*> die Nummer des Datenbankanschlusses.

Beachten Sie, dass die Erstellung von Datenbanktabellen mit Hilfe des Merant-Treibers im EJB-Server-Konfigurations-Browser nicht unterstützt wird. Zum Erstellen von Datenbanktabellen mit dem Merant-Treiber müssen Sie statt dessen den Schema-Browser verwenden.

VisualAge(R) für Java(TM)

Version 4.0

DB2(R) Stored Procedure Builder (Erstellungsprogramm für gespeicherte Prozeduren)

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einführung
2.0 Voraussetzungen
3.0 Installation des DB2-Erstellungsprogramms für gespeicherte Prozeduren
4.0 Einschränkungen und bekannte Probleme
5.0 Hinweise und Tipps
    5.1 Exportieren einer Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren
6.0 Dokumentation

1.0 Einführung

Mit dem IBM(R) DB2 Erstellungsprogramm für gespeicherte Prozeduren können Sie gespeicherte Prozeduren erstellen, installieren und testen.

2.0 Voraussetzungen

Folgende Komponenten müssen installiert sein, bevor das Erstellungsprogramm für gespeicherte Prozeduren ausgeführt werden kann:

1. IBM DB2 Version 6 oder höher
2. IBM DB2 Software Developer's Kit (SDK) einschließlich der Komponente IBM DB2 Stored Procedure Builder (Erstellungsprogramm für gespeicherte Prozeduren)

Weitere Informationen über die Konfiguration der DB2 Universal Database für das Erstellungsprogramm für gespeicherte Prozeduren finden Sie in den Release-Hinweisen zu DB2.

Wenn Sie im Erstellungsprogramm für gespeicherte Prozeduren statische gespeicherte Prozeduren erstellen oder verwenden müssen, müssen Sie das Projekt SQLJ Runtime Library zu Ihrem Arbeitsbereich hinzufügen.

Führen Sie folgende Schritte aus, um das Projekt SQLJ Runtime Library zum Arbeitsbereich hinzuzufügen:

1. Wählen Sie in der Workbench die Option **Fenster > Repository-Explorer** aus.
2. Wählen Sie **SQLJ Runtime Library** im Teilfenster 'Projektnamen' aus.
3. Wählen Sie eine Edition von SQLJ Runtime Library im Teilfenster 'Edition' aus.
4. Wählen Sie **Editionen - > zum Arbeitsplatz hinzufügen** aus.

Wenn Sie das Erstellungsprogramm für gespeicherte Prozeduren mit DB2-Datenbanken, Version 6 oder höher verwenden, müssen Sie auf Ihrer Maschine Fixpack 2 oder eine höhere Version installiert haben.

3.0 Installation des Erstellungsprogramms für gespeicherte Prozeduren

Um das Erstellungsprogramm für gespeicherte Prozeduren zu installieren, führen Sie das Installationsprogramm von VisualAge für Java 4.0 aus und wählen Sie die Komponente Enhanced Database Support (Erweiterte Datenbankunterstützung) aus. Beim Installieren dieser Komponente wird gleichzeitig das Erstellungsprogramm für gespeicherte Prozeduren installiert.

4.0 Einschränkungen und bekannte Probleme

Dialogfensterinformationen und Einschränkungsprüffenster erscheinen möglicherweise unterhalb der zugehörigen Schnittstellenfenster des Erstellungsprogramms für gespeicherte Prozeduren. Dies ist ein bekanntes Problem, das das Verhalten des Erstellungsprogramms für gespeicherte Prozeduren nicht beeinflußt.

Führen Sie folgende Schritte aus, um die Dialogfensterinformationen und die Einschränkungsprüfung auszuschalten:

1. Wählen Sie im Projektfenster des Erstellungsprogramms für gespeicherte Prozeduren **Datei > Umgebungseigenschaften** aus.
2. Klicken Sie die Registerkarte 'Unterstützung' an.
3. Nehmen Sie die Auswahl für Dialogfensterinformationen für die Markierungsfelder für die Schnittstellensteuerung und für die Diagnose der constraint-Verletzungen zurück.
4. Klicken Sie **OK** an.

Wenn Sie die Online-Hilfe für das Erstellungsprogramm für gespeicherte Prozeduren im Netscape Navigator unter Microsoft(R) Windows(R) 2000 öffnen, erhalten Sie möglicherweise die folgende Fehlermeldung:

```
Problem mit dem Direktaufruf. [helpfile].htm kann nicht geöffnet werden.
```

Führen Sie eine Neuinstallation von Netscape Navigator unter Windows 2000 durch, um dieses Problem zu beheben.

Die Online-Hilfe für das Erstellungsprogramm für gespeicherte Prozeduren funktioniert nicht in Netscape 4.7. Das Hilfeprogramm wird zwar im Browser angezeigt, ist aber von einer Fehlermeldung begleitet, dass der Dateiname "nicht geöffnet werden kann". Dieses Problem findet sich nicht bei Netscape-Browsern aus frühe-

ren oder späteren Versionen als Netscape 4.7. Dieses Problem kann auch in einigen Versionen des Internet Explorers auftreten.

5.0 Hinweise und Tipps

5.1 Exportieren einer Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren

Das Erstellungsprogramm für gespeicherte Prozeduren sichert Verbindungsinformationen und nicht erstellte Objekte von gespeicherten Prozeduren in einer Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren (*.SPP). Diese Projektdatei wird im VisualAge für Java-Projekt gespeichert, das Sie zum Aufbauen des Erstellungsprogramms für gespeicherte Prozeduren verwendet haben.

Da VisualAge für Java Projektdateien des Erstellungsprogramms für gespeicherte Prozeduren als Ressourcen behandelt, müssen Sie sicherstellen, dass Ressourcendateien während des Exports eines VisualAge für Java-Projekts kopiert werden. Wenn Sie ein VisualAge für Java-Projekt in eine jar-Datei oder ein Repository exportieren, ohne die Ressourcendateien zu exportieren, ist die Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren nicht automatisch enthalten.

Führen Sie folgende Schritte aus, um ein VisualAge für Java-Projekt in eine jar-Datei (einschließlich der Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren) zu exportieren:

1. Wählen Sie in der Workbench die Registerkarte 'Projekte' aus.
2. Wählen Sie in der Baumstruktursicht das Projekt aus, das die Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren enthält, die Sie exportieren wollen.
3. Wählen Sie **Datei > Exportieren** aus. Der SmartGuide 'Exportieren' wird geöffnet.
4. Wählen Sie den Radioknopf **Jar-Datei** aus und klicken Sie **Weiter** an.
5. Stellen Sie sicher, dass das Markierungsfeld 'Ressource' ausgewählt ist.
6. Klicken Sie den Knopf **Details** neben dem Markierungsfeld 'Ressource' an.
7. Wählen Sie die Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren (*.SPP) aus der Liste der Ressourcendateien aus. Klicken Sie **OK** an.
8. Klicken Sie **Fertig stellen** an.

Führen Sie folgende Schritte aus, um ein VisualAge für Java-Projekt in ein Repository (einschließlich der Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren) zu exportieren:

1. Wählen Sie in der Workbench die Registerkarte 'Projekte' aus.
2. Wählen Sie in der Baumstruktursicht das Projekt aus, das die Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren enthält, die Sie exportieren wollen.
3. Wählen Sie **Datei > Exportieren** aus. Der SmartGuide 'Exportieren' wird geöffnet.
4. Wählen Sie den Radioknopf **Repository** aus, Klicken Sie **Weiter** an.
5. Wählen Sie ein lokales oder gemeinsam benutztes Repository aus und geben Sie die erforderlichen Repository-Informationen ein.
6. Klicken Sie **Fertigstellen** an.
7. Importieren Sie das in Schritt 5 angegebene Repository in das Zielexemplar von VisualAge für Java.

8. Kopieren Sie in Ihrem Dateisystem die Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren aus dem Unterverzeichnis für die Projektquelle
   x:\ibmvjava\ide\project_resources\ihr_projekt
   in das Unterverzeichnis des Zielprojekts. Dadurch wird sichergestellt, dass die Projektdatei des Erstellungsprogramms für gespeicherte Prozeduren im Zielprojekt wiederhergestellt ist.
   Hinweis: Im Beispiel ist x:\ibmvjava das Laufwerk und Verzeichnis, in dem VisualAge für Java 4.0 installiert ist, und ihr_projekt ist der Name des Projektunterverzeichnisses.

6.0 Dokumentation

Die Konzeptdatei "Overview of debugging stored procedures (Debuggen von gespeicherten Prozeduren - Überblick)" hat sich geändert und wurde nicht übersetzt.

VisualAge(R) für Java(TM)

Version 4.0

SQLJ

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einführung
2.0 Menüpunkte
    2.1 Arbeitsbereich / Tools / SQLJ / Eigenschaften
    2.2 Arbeitsbereich / Tools / SQLJ / Import
    2.3 Tools / SQLJ / Editieren
    2.4 Tools / SQLJ / Umsetzen
    2.5 Tools / SQLJ / Erstellen einer SQLJ Debug-Klassendatei
3.0 Unterstützungsbibliotheken
    3.1 Version
    3.2 Aufbauzeit
    3.3 Laufzeit
4.0 Anpassung
    4.1 Zusätzlicher Klassenpfad
    4.2 Klassenname des Umsetzers
    4.3 Methodenname des Umsetzers
5.0 Bekannte Probleme und Einschränkungen
    5.1 Vorzeitige Prozessbeendigung
    5.2 Fehlende Bibliotheken für dynamische Verbindungen (Dynamic Link Library)
    5.3 Generierungseinschränkungen für Debug-Klassendateien
    5.4 Dateiendzeichen
    5.5 Innere Klassen können keine statischen Felder definieren
6.0 Hinweise und Tipps
    6.1 Anpassung
    6.2 Aktualisierung einer Projektressourcensicht
    6.3 In ein bestehendes Paket importieren
7.0 Fehlernachrichten
    7.1 Importieren
8.0 JDBC(TM)-Treiber

1.0 Einführung

Dateien mit der Erweiterung ″.sqlj″ sind Java-Quellendateien, die Java-Quellencode mit eingebetteten SQL-Anweisungen enthalten. Das SQLJ-Tool stellt die Mittel zur Durchführung des folgenden Prozesses zur Verfügung:

1. Eine ″.sqlj″-Datei wird durch einen als ″Umsetzung″ bezeichneten Prozeß zu einer ″.java″-Datei sowie einer oder mehreren Profildateien mit der Erweiterung ′.ser′ konvertiert. Die eingebetteten SQL-Anweisungen in der ″.sqlj″-Datei werden zu JDBC (TM)-Aufrufen in der generierten ″.java″-Datei konvertiert.
2. Die resultierende ″.java″-Datei wird in ein VisualAge für Java IDE-Projekt importiert.
3. Die originale ″.sqlj″-Datei sowie die resultierenden ″.ser″-Dateien werden in das Verzeichnis ′project_resources′ des Projekts kopiert und in einem ″Projekt-Ressourcen″-Fenster angezeigt.
4. Während der Umsetzung kann ein optionaler Semantik-Check gegen die Referenz-Datenbank durchgeführt werden.

2.0 Menüpunkte

2.1 Arbeitsbereich / Tools / SQLJ / Eigenschaften

Dieses Menü zeigt einen Eigenschaftsdialog an. Folgende Eigenschaften können spezifiziert werden:

1. Die zu verwendende Codierung
2. Durchführung von Online-Semantik-Checks gegen die Datenbank - ja oder nein
3. Der JDBC-Treiber, der für Online-Semantik-Checks verwendet werden soll
4. Die URL der Datenbank für Online-Semantik-Checks
5. Die Benutzer-ID für den Zugang zur Datenbank für Online-Semantik-Checks
6. Das Kennwort für den Zugang zur Datenbank für Online-Semantik-Checks

2.2 Arbeitsbereich / Tools / SQLJ / Import

Dieser Menüpunkt zeigt ein Dialogfenster ′SQLJ Dateiimport′ an. Folgende Eigenschaften können spezifiziert werden:

1. Der Name des Projekts, in das die ″.sqlj″-, ″.java″- und ″.ser″-Dateien importiert werden sollen.
2. Der Name der zu importierenden ″.sqlj″-Datei.
3. Ob ein Umsetzungsprozess durchgeführt werden soll.

Wenn der Umsetzungsprozess inaktiviert wird, wird nur die ″.sqlj″-Datei importiert.

2.3 Tools / SQLJ / Editieren

Dieser Menüpunkt sowie die nachfolgenden zwei Menüpunkte sind Aktionen, die gegen eine zuvor importierte ″.sqlj″-Datei durchgeführt werden können. Die Menüpunkte können durch Klicken mit der rechten Maustaste auf eine ″.sqlj″-Datei auf der Ressourcenseite des Projekt-Browsers aufgerufen werden.

Dieser Menüpunkt zeigt ein Editor-Fenster für eine ″.sqlj″-Datei an. Die Datei kann editiert und gesichert werden.

Beachten Sie, dass der Menüpunkt ″Speichern unter″ im Editor den Inhalt des Editor-Fensters in der angegebenen Datei sichert. Nach Abschluß der Operation ″Speichern unter″ fährt der Editor jedoch fort, die alte Datei zu editieren, was auch durch den Dateinamen in der Titelleiste angegeben wird.

2.4 Tools / SQLJ / Umsetzen

Dieser Menüpunkt führt eine Umsetzung der angegebenen ″.sqlj″-Datei sowie einen optionalen Semantik-Check innerhalb des aktuellen Projekts durch.

2.5 Tools / SQLJ / Erstellen einer SQLJ Debug-Klassendatei

Dieser Menüpunkt erstellt eine Klassendatei und stellt sie in das Verzeichnis 'project_resources', das die ″.sqlj″-Datei enthält. Sie können diese Datei in der Ressourcenseite des Projekt-Browsers anzeigen.

Die generierte Klassendatei enthält Debug-Zeilennummern, die mit den Zeilennummern der ".sqlj"-Datei übereinstimmen, und verweist auf die ".sqlj"-Datei als ihre Quellendatei.

Die Klassendatei und die originale ″.sqlj″-Datei können mit dem Distributed Debugger verwendet werden.

3.0 Unterstützungsbibliotheken

3.1 Version

Die mit VisualAge für Java mitgelieferten SQLJ-Unterstützungsbibliotheken unterstützen Version 1.0.1.1 der SQLJ-Spezifikation.

3.2 Build-Time

Ein mit VisualAge für Java gelieferter Umsetzer befindet sich in der Datei ″sqlj-translator.zip″ innerhalb des Verzeichnisses "x:\vajava\ide\tools\com-ibm-ivj-sqlj\lib", wobei *vajava* Ihr Installationsverzeichnis für VisualAge für Java ist.

3.3 Run-Time

Die generierten ″.java″-Dateien enthalten Aufrufe an die SQLJ-Laufzeit-Bibliothek. Diese Bibliothek wird mit VisualAge für Java in drei Formaten im Verzeichnis ″x:\vajava\eab\runtime30″ mitgeliefert:

1. sqlj-runtime.jar ist eine komprimierte Version der Laufzeit-Bibliothek zur Anwendung mit Applets
2. sqlj-runtime.zip ist eine unkomprimierte Version der Laufzeitbibliothek

4.0 Anpassung

Eine SQLJ-Tool-Anpassungsdatei wird bereitgestellt, so dass ein spezifischer Umsetzer mit dem SQLJ-Tool verwendet werden kann. Diese Datei heißt "SQLJ-SupportToolTranslator.properties" und befindet sich im Verzeichnis "x:\vajava\ide\tools\com-ibm-ivj-sqlj". Diese Datei hat den folgenden Inhalt:

```
additionalclasspath = lib\\sqljtranslator.zip;x:\\sqllib\\java\\db2java.zip;
translatorclassname = sqlj.tools.Sqlj
translatormethodname = statusMain
```

4.1 Zusätzlicher Klassenpfad

Das SQLJ-Tool hängt den angegebenen Klassenpfad (Classpath) an den vorhandenen Klassenpfad an, bevor das Tool versucht, nach der Umsetzerklasse zu suchen.

4.2 Klassenname des Umsetzers

Das SQLJ-Tool geht davon aus, dass der angegebene Klassenname die Klasse ist, die den Umsetzungsprozeß der Konversion von ".sqlj"-Dateien in ".java"- und ".ser"-Dateien ausführt. Der Standardklassenname ist der Name einer Klasse, die sich in der sqlj-translator.zip-Datei befindet, die im Paket von VisualAge für Java enthalten ist.

4.3 Methodenname des Umsetzers

Das SQLJ-Tool ruft die angegebene Methode der Klasse dynamisch auf, die durch die Umsetzerklassenname-Spezifikation (translatorclassname) angegeben wurde, um die Umsetzung auszuführen.

5.0 Bekannte Probleme und Einschränkungen

5.1 Vorzeitige Prozessbeendigung

Einige Umsetzer und Anpassungsprogramme behandeln Fehler, indem Sie eine Nachricht an stdout senden und anschließend den Umsetzungsprozess beenden. Wenn dieser Fall eintritt, wird der Prozess vorzeitig beendet, und es wird keine Abschlußnachricht angezeigt.

5.2 Fehlende Bibliotheken für dynamische Verbindungen (Dynamic Link Library)

Wenn eine Bibliothek für dynamisches Verbinden eines Umsetzers fehlt, wird möglicherweise eine Nachricht wie die folgende generiert: "java.lang.UnsatisfiedLinkError: kein db2sqlj im Pfad der gemeinsam benutzten Bibliothek". Der Umsetzungsprozess wird möglicherweise vorzeitig beendet. Wenn der Datenbanktreiber die dem Zugriff auf die Datenbank zugeordneten Bibliotheken nicht finden kann, berichtet der Umsetzer, dass er den Treiber nicht laden kann.

5.3 Generierungseinschränkungen für Debug-Klassendateien

Die Erstellung einer Debug-Klassendatei ist folgendermaßen eingeschränkt:

1. Die ".sqlj"-Datei ist kleiner als die generierte ".java"-Datei.
2. SQLJ-Anweisungen werden an der Position der SQLJ-Anweisung erweitert, und der Umsetzer nimmt keine Optimierung der Codeposition vor.

5.4 Dateiendezeichen

Wenn eine SQLJ-Datei ein Dateiendezeichen wie hex 1a an ihrem Ende aufweist, schlägt die Umsetzung fehl. SQLJ-Dateien werden auf dieselbe Weise wie Java-Dateien behandelt, wo eine javac-Kompilierung einer Java-Datei, die ein Dateiendezeichen enthält, ebenfalls fehlschlägt.

5.5 Innere Klassen können keine statischen Felder definieren

Eine Klasse, die über statische Methoden und Variablen verfügt, ist in einer inneren Klasse durch den SQLJ-Umsetzer nicht zulässig. Stellen Sie sicher, dass die

Deklaration der Klasse mit statischen Methoden und Variablen einer Klassendeklaration vorangeht. Das Stellen einer Klasse, die statische Felder hat, innerhalb einer inneren Klasse resultiert in einem Kompilierfehler.

6.0 Hinweise und Tipps

6.1 Anpassung

Führen Sie folgende Schritte aus, bevor Sie das SQLJ-Tool verwenden:

1. Wählen Sie in der Workbench 'Arbeitsbereich - Tools - SQLJ - Eigenschaften' aus und geben Sie die Umsetzer- und Datenbankoptionen an.

6.2 Aktualisierung einer Projektressourcensicht

Wählen Sie von der Sicht 'Projektressourcen' die Option 'Fenster - Aktualisieren' aus, um die Sicht zu aktualisieren. Sie müssen das Fenster aktualisieren, um Klassendateien anzuzeigen, die generiert werden, wenn Sie eine SQLJ-Datei importieren oder umsetzen oder wenn Sie eine SQLJ-Debug-Klassendatei erstellen.

6.3 In ein bestehendes Paket importieren

Wenn der Paketname in einer SQLJ-Datei, die Sie importieren, identisch mit dem Paketnamen ist, der in einem anderen Projekt als dem ausgewählten Projekt existiert, werden alle Dateien in das andere Projekt importiert. Hierbei wird eine Warnung angezeigt mit dem Namen des Projekts, in das die SQLJ-Datei importiert wird.

Der Import kann ebenfalls fehlschlagen, wenn keine ausreichende Berechtigung zum Importieren in das angegebene Projekt besteht.

7.0 Import-Fehlernachrichten

Beim Importieren einer SQLJ-Datei in ein Projekt oder beim Umsetzen von Dateien, die als Ressourcen in ein Projekt importiert werden, kann ein Fehler auftreten, wenn das Projekt ein gemeinsam benutztes Projekt ist und nicht Eigentum des aktuellen Arbeitsbereichsbenutzers ist. Dies kann geschehen, da das Projekt ein gemeinsames Projekt ist und nicht dem Benutzer des aktuellen Arbeitsbereichs gehört.

8.0 JDBC-Treiber

Standardeinstellung: sun.jdbc.odbc.JdbcOdbcDriver
IBM DB2: COM.ibm.db2.jdbc.app.DB2Driver
Oracle : oracle.jdbc.driver.OracleDriver

VisualAge(R) für Java(TM)

Version 4.0

IBM Distributed Debugger und Object Level Trace (OLT)

Version 9.1

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einführung
2.0 Debugger für Workstation-Dokumentation und F1-Hilfe

1.0 Einführung

Diese Release-Hinweise enthalten Dokumentationsinformationen zu VisualAge für Java Distributed Debugger für Workstations. Informationen zu den Voraussetzungen und Einschränkungen für das Produkt enthalten die Release-Hinweise zu IBM Distributed Debugger und Object Level Trace, die mit dem IBM Distributed Debugger geliefert werden.

2.0 Dokumentation

Es folgt eine Liste der Änderungen in der Online-Information, die nicht übersetzt worden sind. Diese Informationen stehen in der englischen PDF-Datei ″Distributed Debugger for Workstation″ zur Verfügung, die im “pdf”-Verzeichnis auf der Produkt-CD zu finden ist.

1. Die folgenden Konzepte wurden seit der letzten Übersetzung der Hilfe von Distributed Debugger hinzugefügt:
   - “Einschränkungen beim Postmortem-Debugging” (Limitations during postmortem debugging)
   - “Postmortem-Debugging unter OS/2” (Postmortem debugging on OS/2)
   - “Postmortem-Debugging unter AIX” (Postmortem debugging on AIX)
   - “Stufen von Postmortem-Debugging” (Stages of postmortem debugging)
   - “Debug-Informationen trennen” (Separate debug information)
   - “Ungewöhnliches Debugger-Verhalten” (Unusual debugger behavior)
   - “Fehler beim Postmortem-Debugging auf einer UNIX-Workstation” (Errors during UNIX workstation postmortem debugging)
2. Die folgenden Aufgaben wurden hinzugefügt:
   - “Einen Prüfpunkt setzen” (Setting a watchpoint)
   - “Speicherauszugsdateien debuggen” (Debugging dump files)
   - “Eine Speicherauszugsdatei abrufen” (Obtaining a dump file)
   - “Den Debugger zum fernen Debuggen von interpretiertem Java starten” (Starting the debugger for debugging interpreted Java remotely)
3. Die folgenden Referenzen wurden hinzugefügt:
   - “Programmprofile” (Program profiles)
   - “Unterstützte Datentypen für Java-Programme” (Supported data types for Java programs)
   - “DER_DBG_DEEP_STEP_DEBUG”
   - “OnException | OffException”
   - “@ITRAPPER; starten” (Start @ITRAPPER;)
   - “OutputPath | Path”
   - “Parameterdatei Trapper” (Trapper parameter file)
   - “DumpNumber | Number”
   - “LogFile”
   - “TraceChild”
   - “UserExit”

4. Die folgenden Konzepte wurden geändert:
   - "Distributed Debugger: Übersicht" (Distributed Debugger: Overview)
   - "Distriubuted Debugger: Überwachungen" (Distributed Debugger: Monitors) heißt nun "Distributed Debugger: Werteteilfenster" (Distributed Debugger: Value Panes), und der Inhalt wurde geändert.
   - "Distributed Debugger: Teilfenster" (Distributed Debugger: Panes) heißt nun "Distributed Debugger: Quellen- und Steuerungsteilfenster" (Distributed Debugger: Source and Control Panes), und der Inhalt wurde geändert.
   - "Ausnahmebedingungsbehandlung" (Exception Handling)
   - "Unterbrechungspunkte" (Breakpoints)
   - "Bearbeitung von exec()" (exec() handling)
   - "Bearbeitung von fork()" (fork() handling)
   - "Zeitpunkt des Zuordnens" (When to attach)
   - "Fernes Debuggen" (Remote debugging)
   - "Bearbeitung von system()" (system() handling)
5. Die Debugger-Task-Dateien, die Informationen über folgende Themen enthalten, wurden grundlegend geändert:
   - Zuordnen, Starten und erneut Starten
   - Arbeiten mit Debugger-Werteteilfenstern (früher 'Überwachungen') sowie Quellen- und Steuerungsteilfenstern (früher 'Teilfenster')

   Die übrigen Task-Dateien weisen geringfügige Änderungen auf.
6. Alle Debugger-Referenzdateien zu Befehlen und Umgebungsvariablen wurden umfassend geändert. Die übrigen Referenzdateien weisen geringfügige Änderungen auf.
7. Die folgenden Tasks sind seit der letzten Übersetzung der Hilfe von Distributed Debugger weggefallen:
   - "PM-Anwendungen debuggen" (Debugging PM Applications)
   - "Eine Klasse oder ein Paket registrieren" (Registering a class or package)
   - "Den Debugger unter HP-UX oder Solaris starten" (Starting the debugger on HP-UX or Solaris)
   - "Den Debugger unter Windows 95 oder Windows 98 starten" (Starting the debugger on Windows 95 or Windows 98)
8. Die folgenden Referenzen sind seit der letzten Übersetzung der Hilfe von Distributed Debugger weggefallen:
   - "Befehl irmtdbg " (irmtdbg command)
   - "Compileroptionen für High Performance Compiled Java - " (High Performance Compiled Java compiler options)
   - "Debug-Einschränkungen für interpretiertes Java - " (Interpreted Java debug limitations)
   - "Umgebungsvariable DER_DBG_LOCAL_PATH" (DER_DBG_LOCAL_PATH environment variable)
   - "Umgebungsvariable DER_DBG_NUMBEROFELEMENTS" (DER_DBG_NUMBEROFELEMENTS environment variable)
   - "Debug-Einschränkungen für Visual C++" (Visual C++ debug limitations)

VisualAge(R) für Java(TM)

Version 4.0

Externe Versionssteuerung

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einführung
2.0 Unterstützte SCM-Systeme
3.0 Microsoft Visual SourceSafe
4.0 VisualAge TeamConnection
5.0 Rational ClearCase
6.0 PVCS Version Manager and Dimensions
7.0 StarTeam Information
8.0 Korrekturen zur Dokumentation
    8.1 Unterbrochene Verknüpfungen im Navigationsteilfenster
    8.2 Änderungen in den Online-Informationen, die nicht übersetzt worden sind

1.0 Einführung

Die VisualAge für Java IDE bietet eine Versionssteuerungs-Schnittstelle zur Überprüfung von Dateien innerhalb und außerhalb eines externen Software-Konfigurations-Management (SCM)-Systems an.

2.0 Unterstützte SCM-Systeme

Die Schnittstelle von VisualAge für Java zu externen Versionssteuerungssystemen verwendet die Source Code Control-API (SCC-API) von Microsoft. Sie unterstützt die folgenden SCM-Systeme:

- ClearCase für Windows NT(R), Version 4.0, der Rational Software Corporation
- PVCS Version Manager, Version 6.6 und 6.7, von MERANT
- VisualAge TeamConnection(TM), Version 3.03, der IBM Corporation
- Visual SourceSafe(TM), Version 6.0, von Microsoft(R)

Andere SCM-Systeme, die mit der SCC-Schnittstelle kompatibel sind, sollten ebenfalls funktionieren, doch sind die Tests dieser Systeme von IBM noch nicht vollständig abgeschlossen. Dies schließt unter anderem die folgenden Produkte ein:

- CCC/Harvest von Computer Associates International, Inc
- CM Synergy von Telelogic (früher Continuus CM von Continuus Software)
- CS-RCS / CS-RCS PRO von ComponentSoftware.com
- Perforce von Perforce Software
- PVCS Dimensions von MERANT
- StarTeam von Starbase Corporation

3.0 Microsoft Visual SourceSafe

Wenn Sie mit SourceSafe arbeiten, müssen Sie stets die Dateien aktualisieren, die von Ihrem lokalen Arbeitsverzeichnis zugeordnet werden, anstatt der Dateien, die vom Server zugeordnet werden. Dies liegt an bestimmten Einschränkungen in der SourceSafe-Implementierung der SCC-Schnittstelle.

Damit Ihre Dateien ordnungsgemäß aktualisiert werden, müssen Sie vor dem Aktualisieren Ihr Arbeitsverzeichnis mit dem Inhalt des SourceSafe-Projekts synchronisieren. Hierbei handelt es sich erwartungsgemäß um eine permanente Einschränkung, da Microsoft nicht beabsichtigt, weitere Änderungen an dieser Schnittstelle vorzunehmen.

Führen Sie folgende Schritte aus, wenn Sie in VisualAge für Java mit SourceSafe-Dateien arbeiten:

1. Erstellen Sie in SourceSafe ein Projekt, und ordnen Sie ihm ein Arbeitsverzeichnis zu.
2. Fügen Sie in VisualAge für Java das entsprechende Projekt einer Versionssteuerung zu. Das Standardarbeitsverzeichnis ist möglicherweise nicht richtig eingestellt. Stellen Sie sicher, dass es dem in Schritt 1 ausgewählten Arbeitsverzeichnis entspricht.
3. Wenn Sie Ihre Dateien aktualisieren wollen, müssen Sie jedesmal die neueste Version der Dateien aus SourceSafe in Ihr Arbeitsverzeichnis kopieren. Verwenden Sie hierzu die Operation zum Abrufen der neuesten Version (″Get latest version″) in SourceSafe.

   Wenn Sie versuchen, Ihre Dateien zu aktualisieren, wird ein Dialogfenster mit einer Warnung aufgerufen, um Sie daran zu erinnern, vor dem Aktualisieren Ihrer Dateien zunächst das entsprechende Arbeitsverzeichnis zu aktualisieren. Sie können SourceSafe über dieses Dialogfenster aufrufen.

4.0 VisualAge TeamConnection

Sie müssen Version 3.03 von VisualAge TeamConnection(TM) installiert haben, wenn Sie den Team Connection Handler verwenden wollen.

5.0 Rational ClearCase

Beim Hinzufügen eines Projekts zur Versionssteuerung müssen Sie die folgenden Schritte zur Einstellung der Option “Neue Projektdateien suchen” ausführen:

1. Wählen Sie das Projekt aus, das Sie zur Versionssteuerung hinzufügen möchten. Wählen Sie anschließend aus dessen Kontextmenü die Optionen **Tools > Externe Versionssteuerung > Zur Versionssteuerung hinzufügen** aus.
2. Wählen Sie **Microsoft SCCI** aus und klicken Sie **OK** an.
3. Wählen Sie **ClearCase** und anschließend das Markierungsfeld **Fortgeschrittene Optionen zeigen, sofern verfügbar** aus. Klicken Sie **OK** an.
4. Klicken Sie die Registerkarte **Allgemein** an. Wählen Sie anschließend das Markierungsfeld **Neue Projektdateien suchen** aus.
5. Schließen Sie das Dialogfenster, um mit dem Hinzufügen Ihres Projekts zur Versionssteuerung fortzufahren.

Projekte, die mit einer dynamischen Ansicht verbunden sind, müssen vom Arbeitsverzeichnis aus aktualisiert werden.

1. Wählen Sie das Projekt aus, das Sie aktualisieren möchten, und wählen Sie aus dessen Kontextmenü die Optionen **Tools > Externe Versionssteuerung> Eigenschaften** aus.
2. Klicken Sie die Indexzunge **Microsoft SCCI** an.
3. Wählen Sie den Radioknopf **Projekt aus dem Arbeitsverzeichnis aktualisieren** aus. Klicken Sie **OK** an.

Bei weiteren Fragen zur Integration von ClearCase mit VisualAge für Java ziehen Sie bitte die Dokumentation ″ClearCase VisualAge für Java Integration″ heran, die von Rational mit einer Programmkorrektur für ClearCase 4.0 und höhere Versionen geliefert wird.

6.0 PVCS Version Manager and Dimensions

Wenn Sie PVCS Dimensions 6.0 oder die PVCS Versions-Manager-Entwicklungsschnittstelle, Versionen 6.6.01 oder 6.6.10 verwenden, können Sie aktuelle Informationen zum Arbeiten mit VisualAge für Java 3.5 der MERANT KnowledgeBase entnehmen. Gehen Sie zur Web-Site http://support.merant.com/ und klicken Sie das KnowledgeBase-Link an. Wählen Sie auf der nächsten Seite das PVCS-Link aus. Geben Sie anschließend “VisualAge für Java 3.5” in das Suchfeld ein und wählend Sie die Option ″Boolesche Suche″ aus.

7.0 StarTeam-Informationen

Um StarTeam mit VisualAge für Java zu verwenden, müssen Sie Unterstützung für die SCC-Schnittstelle installiert haben. Wählen Sie die Option “Angepasst” in der StarTeam Client-Installation aus. Wählen Sie anschließend “Entwickler-Studio-Integration” aus, die die SCC-kompatible Integration installiert.

8.0 Dokumentation

8.1 Unterbrochene Verknüpfungen im Navigationsteilfenster

In den NL-Editionen von VisualAge für Java, Professional Edition, funktioniert im Navigationsteilfenster die folgende Verknüpfung nicht:

- Entwicklung im Team (eine Verknüpfung in der Konzept-Hilfe)

Diese Verknüpfung mit Informationen zur Entwicklung im Team funktioniert nicht, da die Team-Komponente nicht in der Professional Edition von VisualAge für Java enthalten ist. Keine der in dieser Datei enthaltenen Informationen ist für die Entwicklung in der Professional Edition von VisualAge für Java von Belang.

8.2 Änderungen in den Online-Informationen, die nicht übersetzt worden sind

Es folgt eine Liste der kritischen Änderungen an den Online-Informationen, die nicht übersetzt wurden. Die neuesten Versionen der Konzept- und Task-Dateien stehen in der englischen PDF-Datei im Verzeichnis “pdf” auf der Produkt-CD zur Verfügung.

- Die Konzeptdatei “Schnittstelle zu externen Versionssteuerungssystemen” (Interface to external version control systems) wurde geändert.
- Die Symbole für den Status Ihres Projekts wurden seit Version 3.5 in der GUI geändert. Diese Änderung wurde nicht in die Task-Dateien übernommen.

- Einige der Kontext-Hilfedateien enthalten Feldüberschriften, die nicht mit der GUI übereinstimmen. Die Funktionalität bleibt jedoch wie in der Datei beschrieben erhalten.

VisualAge(R) für Java(TM)

Version 4.0

JavaServer Pages (JSP)/Servlet-Entwicklungsumgebung (Development Environment)

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einführung
2.0 Voraussetzungen
    2.1 Software-Voraussetzungen
    2.2 JSP-Unterstützung
3.0 Einschränkungen und bekannte Probleme
    3.1 Beim Laden von JSP-Dateien wird ein “Klasse wurde nicht gefunden”-Fehler angezeigt
    3.2 Die Fehlernachricht “Seitenkompilierte Klasse kann nicht geladen werden” erscheint im Browser
    3.3 Nach dem Ändern von JSP-Dateien werden Ausnahmebedingungen im Browser angezeigt
    3.4 Beim Starten der Servlet-Engine erscheint der Fehler “Java.io:ClassNotFoundDef”
    3.5 Symbole im JSP Execution Monitor werden nicht ordnungsgemäß neu angezeigt
    3.6 JSP-Dateien können nicht in einem Verzeichnis gesichert werden, dessen Name einen Punkt (“.”) enthält
    3.7 Internet Explorer ändert manchmal die Anforderungs-URL
    3.8 “Quelle nicht verfügbar” beim Debuggen einer JSP-Datei mit dem IDE-Debugger
    3.9 Importiertes Paket während des Kompilierens nicht gefunden
    3.10 Der JSP Execution Monitor kann nach Absturz von VisualAge für Java nicht gestartet werden
    3.11 Probleme beim Bedienen einer importierten JSP-Datei
    3.12 Einschränkungen beim JSP-Verzeichnisnamen
    3.13 Probleme bei den Knöpfen Step/Ausführen/Schnell vorwärts im JSP Execution Monitor
    3.14 Einschränkungen beim gleichzeitigen Einsatz von JSP Execution Monitor und IDE-Debugger
    3.15 Einschränkungen bei Unterbrechungspunkten bei JSP Execution Monitor in JSP 1.1
    3.16 Starten von JSP Execution Monitor schlägt fehl
    3.17 Servlet-Engine-Modus auf Full 2.2 Compliance Mode ändern
    3.18 Servlets, EJBs und CORBA.COMM_FAILURE
    3.19 Java-Beans in Standardpaketen können nicht in Servlets importiert werden
4.0 Dokumentation

1.0 Einführung

Diese Release-Hinweise enthalten Informationen über verschiedene Einschränkungen und bekannte Probleme beim IBM(R) WebSphere(TM) Test Environment. Dieses Dokuemt enthält außerdem Software-Voraussetzungen sowie Hinweise, Tipps und Lösungen zur Problemumgehung.

Folgende Ändeurngen zwischen VisualAge für Java Version 3.5 und Version 3.5.3 bestehen:

- JSP 1.1-Unterstützung
- Der Knopf **Mit Step into in IDE-Debugger eintreten** in der JSP-Ausführungsüberwachung (JSP Execution Monitor), mit dem Sie unter Verwendung des IDE-Debbugers den JSP-Quellcode durchgehen können.

2.0 Voraussetzungen

2.1 Software-Voraussetzungen

Um mit den JavaServer-Seiten in IBM VisualAge für Java arbeiten zu können, müssen Sie zunächst die erforderlichen Funktionen aus dem Repository in den Arbeitsbereich laden.

Gehen Sie folgendermaßen vor, um eine Feature zu installieren:

1. Wählen Sie **Datei -> Schnellstart -> Funktionen -> Funktionen hinzufügen** aus.
2. Wählen Sie anschließend **IBM WebSphere-Testumgebung** aus.
3. klicken Sie **OK** an.

Die folgenden Projekte werden automatisch in Ihren IDE-Arbeitsbereich geladen:

- IBM WebSphere-Testumgebung
- IBM JSP-Beispiele

Die IBM WebSphere-Testumgebungs-Funktion enthält die WebSphere-Integrationskomponenten, die Servlet-API-Klassen (Application Programming Interface) und den JSP Execution Monitor.

Zusätzliche Informationen zum Arbeiten mit JSPs und Servelets in VisualAge für Java finden Sie in der Online-Hilfe.

2.2 JSP-Unterstützung

VisualAge für Java, Version 4.0, unterstützt die Spezifikationen von JSP 0.91, JSP 1.0 und JSP 1.1. Die Standardspezifikation lautet JSP 1.0. Ausführliche Anweisungen zum Wechseln zwischen den JSP-Jonfigurationsstufen finden Sie in der Online-Hilfe.

3.0 Einschränkungen und bekannte Probleme

3.1 Beim Laden von JSP-Dateien wird ein "Klasse wurde nicht gefunden"-Fehler angezeigt

Um eine Klasse in VisualAge für Java ausführen zu können, auf die eine JSP-Datei zeigt, muss sich die Klasse innerhalb der IDE befinden. Sie müssen das Projekt, das die Klasse enthält, zu Ihrem IDE-Klassenpfad hinzufügen.

Hinweis: Klassen können sich in .jar-Dateien befinden. In diesem Fall muss der Klassenpfad auf die .jar-Datei selbst zeigen.

3.2 Die Fehlernachricht "Seitenkompilierte Klasse kann nicht geladen werden" erscheint im Browser

Dieses Problem kann folgende Ursache haben:

Wenn Sie eine Änderung der Klassenpfadeigenschaft vornehmen und sie anwenden, BEVOR Sie Ihre erste JSP-Seite laden (d. h. bevor das Projekt 'JSP Page Compile Generated Code' erstellt wird), wird die vorinstallierte Klassenpfadreferenz entfernt. Wenn Sie dann versuchen, eine JSP-Seite zu laden (wodurch das Projekt 'JSP Page Compile Generated Code' erstellt wird), weiß der Server nicht, wo der generierte Code zu finden ist, und kann folglich die kompilierte Seite nicht abrufen. Im Browser wird die Ausnahme-Nachricht "Seitenkompilierte Klasse kann nicht geladen werden" (Cannot Load Page Compiled Class) ausgegeben.

Wenn diese Ausnahmebedingung auftritt, nachdem der Klassenpfad geändert wurde, können Sie das Problem lösen, indem Sie die folgenden Schritte ausführen:

1. Schließen Sie die WTE-Servlet-Engine.
2. Fügen Sie 'JSP Page Compile Generated Code' zum Projektklassenpfad hinzu.
3. Starten Sie die Servlet-Engine erneut.
4. Bedienen Sie die JSP-Seite erneut.

3.3 Nach dem Ändern von JSP-Dateien werden Ausnahmebedingungen im Browser angezeigt

Wenn in Ihrem Browser nach dem Laden von JSP-Dateien wiederholt unerwartete Ausnahmebedingungen auftreten, müssen Sie möglicherweise den Seitenkompilierprozess dazu zwingen, eine JSP-generierte Quelle erneut zu erstellen, und anschließend die neue Version in VisualAge für Java importieren lassen. Führen Sie hierzu folgende Schritte aus:

1. Schließen Sie die Servlet-Engine.
2. Löschen Sie alle Pakete aus dem Projekt 'JSP Page Compile Generated Code', belassen Sie das Projekt selbst jedoch im Arbeitsbereich.
3. **Für die JSP 0.91-Spezifikation:** Löschen Sie das gesamte Verzeichnis, in das der Seiten-Compiler die generierte Quelle schreibt:
   `X:\VisualAge for Java\ide\project_resources\IBM WebSphere Test Environment\temp\default_app\pagecompile`
   Hierbei ist *X:\VisualAge for Java* das Installationsverzeichnis von VisualAge für Java.

   Für die JSP 1.0-Spezifikation: Löschen Sie alle Dateien und deren Unterverzeichnis, in das der Seiten-Compiler die generierte Quelle schreibt:
   `X:\VisualAge for Java\ide\project_resources\IBM WebSphere Test Environment\temp\JSP1_0\default_app`
   Hierbei ist *X:\VisualAge for Java* das Installationsverzeichnis von VisualAge für Java.
4. Starten Sie die Servlet-Engine erneut.
5. Bedienen Sie Ihre JSP-Seiten erneut. Daraufhin wird der neu generierte Code importiert.

3.4 Beim Starten der Servlet-Engine erscheint der Fehler "Java.io:ClassNotFound-Def"

Gelegentlich gibt die Servlet-Engine beim Starten die oben genannte Ausnahmebedingung aus. Wenn dies eintritt, wird der Debugger aufgerufen. Um dieses Problem zu lösen, schließen Sie den Debugger, beenden Sie die Servlet-Engine und rufen Sie servlet anschließend erneut auf. WebSphere korrigiert den Fehler. Die Korrektur wird beim nächsten Starten der Servlet-Engine wirksam.

3.5 Symbole im JSP Execution Monitor werden nicht ordnungsgemäß neu angezeigt

Beim doppelten Anklicken der JSP-Quellen-Indexzunge im JSP Execution Monitor wird das JSP-Quellen-Fenster abgehängt und die Symbole im JSP Execution Monitor werden nicht wieder neu aufgebaut. Um dieses Problem zu umgehen, schließen Sie das abgehängte JSP-Quellen-Fenster.

3.6 JSP-Dateien können nicht in einem Verzeichnis gesichert werden, dessen Name einen Punkt enthält

Wenn Sie eine JSP-Datei in einem Verzeichnis sichern, das einen Punkt enthält (beispielsweise "JSP1.0"), zeigt der Browser einen Kompilationsfehler an. Dies ist eine Einschränkung der WebSphere-Testumgebung.

3.7 Internet Explorer ändert manchmal die Anforderungs-URL

Unter Windows(R) NT und Windows 2000 ändern einige Versionen des Internet Explorers die Abfrage-URL automatisch von http://localhost:8080/... in http://local:8080/... um. Wenn dies geschieht, kann der Server die angeforderte Datei nicht finden. Sollte dieses Problem auftreten, empfiehlt sich die Verwendung des Netscape-Browsers.

3.8 "Quelle nicht verfügbar" beim Debuggen einer JSP-Datei mit dem IDE-Debugger

Wenn Sie eine JSP-Datei debuggen (mit dem IDE-Debugger), die eine JSP 1.0-Engine mit dem Markierungsfeld **Generiertes Servlet extern laden** im aktivierten Zustand verwendet, erscheint im Teilfenster des IDE-Debuggers die Nachricht "Quelle nicht verfügbar", wenn sich die JSP-Datei direkt im Dokumentstammverzeichnis befindet. Dies bedeutet, dass sich die JSP-Datei nicht in einem Unterverzeichnis des Dokumentstammverzeichnisses befindet.

Um dieses Problem zu umgehen, fügen Sie das JSP 1.0-Cache-Verzeichnis zum Quellenpfad des Debuggers hinzu. Gehen Sie wie folgt vor, um den Quellenpfad für das Debugging zu setzen:

1. Wählen Sie die Option **Fenster** -> **Optionen** aus.
2. Wählen Sie im Teilfenster **Kodierung -> Debugging** aus.
3. Wählen Sie den Knopf **Editieren** und fügen Sie das JSP-Cache-Verzeichnis hinzu.
4. Das Standardverzeichnis für den JSP 1.0-Cache ist:
   X:\VisualAge for Java/ide/project_resources/IBM WebSphere Test Environment/temp/JSP1_0/default_app
   Hierbei ist *X:\VisualAge for Java* das Installationsverzeichnis.

3.9 Importiertes Paket während des Kompilierens nicht gefunden

Wenn in der JSP 0.91-Unterstützung eine JSP-Datei den Import-Befehl zum Importieren eines Pakets verwendet, während das Markierungsfeld **Syntaxfehlerinformationen abrufen** aktiviert ist, wird der Seiten-Compiler angeben, dass er das importierte Paket während des Kompilierens nicht finden konnte.

Um dieses Problem zu umgehen, erstellen Sie ein leeres Unterverzeichnis mit dem Namen des Pakets im Verzeichnis *X:\VisualAge for Java*/ide/project_resources/IBM WebSphere Test Environment/temp/temp_dir.

Falls das Verzeichnis temp_dir noch nicht vorhanden ist, müssen Sie es zuvor erstellen. Beispielsweise müsste für ein zu importierendes Paket mit dme Namen com.ibm.jsp das Verzeichnis *X:\VisualAge for Java*/ide/project_resources/IBM WebSphere Test Environment/temp/temp_dir/com/ibm/jsp erstellt werden.

3.10 Der JSP Execution Monitor kann nach Absturz von VisualAge für Java nicht gestartet werden

Wenn der JSP Execution Monitor (JSP-Ausführungsüberwachung) beim Absturz von VisualAge für Java eingeschaltet war, löschen Sie die Datei 'monitorStatus.properties' im Verzeichnis *X:\VisualAge for Java*\ide\project_resources\IBM WebSphere Test Environment\com\ibm\ivj\jsp\ui, und versuchen Sie es erneut. Diese Datei zeigt an, ob der JSP Execution Monitor eingeschaltet (true) oder ausgeschaltet (false) ist. War der JSP Execution Monitor ausgeschaltet, obwohl die Datei angab, dass der Monitor eingeschaltet war, tritt eine Ausnahmebedingung auf, da der JSP Execution Monitor nicht gefunden werden kann.

3.11 Probleme beim Bedienen einer importierten JSP-Datei

Wenn Probleme beim Bedienen einer JSP-Datei auftreten, die mit ausgewähltem Markierungsfeld **Generiertes Servlet extern laden** importiert wurde, müssen Sie vor dem Importieren des Servlet-Codes sicherstellen, dass dieses Markierungsfeld nicht ausgewählt ist.

3.12 Einschränkungen beim JSP-Verzeichnisnamen

Bei JSP 0.91 und JSP 1.0 muss der Name des Verzeichnisses, das die JSP-Dateien enthält, den Namenskonventionen für Java-Klassen entsprechen. So darf ein Verzeichnisname beispielsweise nicht mit einer Zahl beginnen.

3.13 Probleme bei den Knöpfen 'Step'/'Ausführen'/'Schnell' im JSP Execution Monitor

Manchmal folgt auf das Drücken der Knöpfe 'Step', 'Ausführen' oder 'Schnell' im JSP Execution Monitor keine Reaktion. Drücken Sie in einem solchen Fall den Knopf**Beenden**, damit die JSP richtig ausgeführt wird. Die Ergebnisse werden daraufhin korrekt im Browser angezeigt.

3.14 Einschränkungen beim gleichzeitigen Einsatz von JSP Execution Monitor und IDE-Debugger

Der JSP Execution Monitor reagiert auf keinen Knopf mit Ausnahme des Knopfs **Beenden**, wenn Sie eine JSP-Datei mit Hilfe von IDE-Debugger debuggen. Diese Situation tritt in folgenden Fällen auf:

- Die JSP-Datei wird bedient oder geladen, wobei das Markierungsfeld **Am Anfang der Servicemethode stoppen** im Teilfenster 'Servlet-Engine' des Kontrollzentrums der WebSphere-Testumgebung ausgewählt ist.
- Im generierten Servlet wurde ein Unterbrechungspunkt gesetzt.
- Der Knopf **Mit Step Into in IDE-Debugger eintreten** wurde gedrückt.

3.15 Einschränkungen bei Unterbrechungspunkten bei JSP Execution Monitor in JSP 1.1

Beim Ausführen einer JSP-Datei mit dem Tag *tsx:repeat* im JSP Execution Monitor unter Verwendung von JSP 1.1 stoppt der Monitor bei keinem der Unterbrechungspunkte, die im HTML-Code des Wiederholungsblocks gesetzt sind. Die Unterbrechungspunkte funktionieren nur bis zum letzten Scriptlet im Wiederholungsblock.

3.16 Starten von JSP Execution Monitor schlägt fehl

Wenn das Starten von JSP Execution Monitor fehlschlägt, können Sie das Problem möglicherweise folgendermaßen umgehen:

Überprüfen Sie die Datei MonitorStatus.properties im Verzeichnis project_resources\IBM WebSphere Test Environment\com\ibm\ivj\jsp\ui\, und stellen Sie sicher, dass die Markierung IsMonitorWindowExists auf den Wert ″false″ gesetzt ist. Ist dieser Wert auf ″true″ gesetzt, bearbeiten Sie die Datei und ändern Sie den Wert in ″false″.

3.17 Servlet-Engine-Modus auf Full 2.2 Compliance Mode ändern

Die Servlet-Engine in VisualAge für Java verwendet standardmäßig den Websphere 3.5-Kompatibilitätsmodus. Wenn Sie den Full Servlet 2.2 Compliance Mode haben wollen, erstellen Sie eine property-Eigenschaftsdatei mit dem Namen ivj.serlvet-.properties im Verzeichnis ...\ide\project_resources\IBM Websphere Test Environment\. Diese property-Datei muss den folgenden Inhalt haben:

```
com.ibm.websphere.servlet22.compliant=true
```

Dadurch kann die Servlet-Engine den vollständigen Compliance-Modus verwenden.

3.18 Servlets, EJBs und CORBA.COMM_FAILURE

In IBM WebSphere Application Server Advanced Edition Version 3.5.2 oder höher implementieren JNDI-Kontextobjekte client-seitiges Caching, um die Leistung der JNDI-Suchoperationen zu erhöhen. Gebundene und gesuchte Objekte werden in den Cache gestellt, um nachfolgende Suchvorgänge dieser Objekte zu beschleunigen.

Wenn ein Servlet ein EJB-Home sucht, wird das Home-Objekt in den Cache gestellt. Da das Servlet in der Servlet-Engine läuft, wird der Cache nach der vollständigen Ausführung des Servlets nicht gelöscht, sondern nur dann, wenn die

Servlet-Engine heruntergefahren wird. Daher wird bei der nächsten Ausführung des Servlet das in den Cache gestellte Home-Objekt des Servlets abgerufen.

Wenn der EJB-Server neu gestartet wird, ist das in den Cache gestellte Home-Objekt ″inaktiv″, und das Aufrufen beliebiger Objekte des Objekts führen zu einer Ausnahmebedingung.

Um dieses Verhalten zu vermeiden, können Sie angeben, dass ein JNDI-Objekt kein Caching zum Zeitpunkt seiner Erstellung verwenden soll:

```
Properties p = new Properties();
p.put("com.ibm.websphere.naming.jndicache.cacheobject", "none");
p.put(javax.naming.Context.PROVIDER_URL, url);
p.put(javax.naming.Context.INITIAL_CONTEXT_FACTORY,
com.ibm.ejs.ns.jndi.CNInitialContextFactory");
InitialContext context = new InitialContext(p);
```

Beachten Sie, dass dies Auswirkungen auf die Leistung Ihres Servlets hat; es empfiehlt sich, das Caching wieder zu aktivieren, bevor Sie das Servlet in einer Produktionsumgebung implementieren.

weitere Informationen zum WebSphere JNDI-Caching, finden Sie im Abschnitt 'Programming' (Programmierung) im IBM WebSphere InfoCenter.

3.19 Java-Beans in Standardpaketen können nicht in Servlets importiert werden

Wenn Sie ein Servlet mit dem SmartGuide 'Servlet erstellen' erstellen, können Sie keine Java-Beans im Standardpaket in das Servlet importieren. Wenn Sie dies dennoch tun, erhalten Sie die Fehlernachricht ″Invalid Java bean class“ (″Ungültige Java-Bean-Klasse“).

4.0 Dokumentation

Alle JSP/Servlet-Entwicklungsumgebungs-Dateien wurden seit der letzten Übersetzung geändert. Zusätzlich wurden die folgenden neuen Dateien hinzugefügt:

- ″JSP-Debug-Ablauf″ (JSP debug flow)
- ″Dateinamen für generierte Servlets″ (Generated servlet file names)
- ″Mit WebSphere Studio-Assistenten generierte JSPs/Servlets importieren″ (Importing JSP/Servlet generated by WebSphere Studio wizards)
- ″Hinweise und Tipps″ (Hints and tips)

Die Seite ″Zwischen JSP 1.0- und JSP 0.91-Unterstützung migrieren“ wurde in ″Zwischen JSP-Konfigurationsstufen wechseln“ umbenannt. Der Inhalt dieser Seite wurde geändert.

Die Seite ″JSP 1.0-Programmreferenz“ wurde in ″JSP 1.0- und JSP 1.1-Programmreferenz“ umbenannt. Zwar gibt es keine neuen IBM Erweiterungen, doch wurden einige geringfügige Änderungen am Inhalt dieser Seite vorgenommen.

Die Seite ″JSP-Ausführungsüberwachung verwenden“ wurde umstrukturiert und aktualisiert und enthält jetzt den Knopf **Mit Step into in IDE-Debugger eintreten**.

Die jeweils neuesten Informationen können Sie der englischen Version des Portable Document Format (PDF) für die JSP/Servlet-Entwicklungsumgebung entnehmen.

VisualAge(R) für Java(TM)
Version 4.0

Ferner Zugriff auf die Tool-API

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Einschränkungen und bekannte Probleme
1.1 ExportCodeSpec.setSubdirectories(boolean)
2.0 Hinweise und Tipps
2.1 ToolEnv.connectToWorkspace() returns null

1.0 Einschränkungen und bekannte Probleme

1.1 ExportCodeSpec.setSubdirectories(boolean)

Die Funktionsweise des SmartGuides **Exportieren** in VisualAge für Java wurde geändert. Sie können keine Datei exportieren, ohne die Unterverzeichnisse des Pakets zu generieren, in dem die Datei deklariert ist. Die Tools Integrator API bietet jedoch weiterhin die Möglichkeit, Dateien zu exportieren, ohne diese Unterverzeichnisse zu erstellen. Sie sollten diese API allerdings als 'veraltet' betrachten.

2.0 Hinweise und Tipps

2.1 ToolEnv.connectToWorkspace() returns null

Wenn der Code ToolEnv.connectToWorkspace() null zurückgibt, überprüfen Sie, ob der Klassenpfad der Anwendung sowohl auf die Klassenbibliotheken des IBM IDE-Dienstprogramms als auch auf die lokale Implementierung des IBM IDE-Dienstprogramms unter IDE/project_resources im Dateisystem zeigt.

VisualAge(R) für Java(TM)

Version 4.0

XML Parser für Java

RELEASE-HINWEISE

Inhaltsverzeichnis

1.0 Dokumentation

1.0 Dokumentation

Dieser Abschnitt umfasst Änderungen an der Dokumentation, die nicht übersetzt wurde.

1. Die folgende Task-Information hat sich geändert:
   - "Einen Parser konstruieren" (Constructing a parser)
   - "Die API zur Wiederbestätigung verwenden" (Using the revalidation API)
2. Die Information zur Ausführung der Beispiele hat sich geändert.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Produkte, Programme und Dienstleistungen in dieser Veröffentlichung bedeuten nicht, dass IBM diese in allen Ländern, in denen IBM vertreten ist, anbietet. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können.

Anstelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanfragen sind schriftlich an folgende Adresse zu richten:

IBM Europe
Director of Licensing
92066 Paris La Defense
Cedex,
France

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION STELLT DIESE VERÖFFENTLICHUNG “UNTER ABBEDINGEN DER HAFTUNG FÜR EINEN BESTIMMTEN ZWECK” ZUR VERFÜGUNG, OHNE GEWÄHRLEISTUNGEN JEGLICHER ART, WEDER AUSDRÜCKLICHEN NOCH IMPLIZIERTEN, EINSCHLIESSLICH DER, ABER NICHT BESCHRÄNKT AUF DIE IMPLIZIERTEN GEWÄHRLEISTUNGEN UND BEDINGUNGEN DER NICHT-VERLETZUNG GEWERBLICHER SCHUTZRECHTE, DER MARKTGÄNGIGEN QUALITÄT UND DER EIGNUNG FÜR EINEN BESTIMMTEN ZWECK. In einigen Ländern sind Ablehnungserklärungen bezüglich ausdrücklicher oder implizierter Gewährleistungen nicht zulässig, daher trifft diese Erklärung möglicherweise nicht für Sie zu.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen bzw. neuen Editionen der Veröffentlichung bekanntgegeben. IBM kann jederzeit ohne vorherige Ankündigung Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in dieser Veröffentlichung auf Web-Sites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Web-Sites dar. Das über diese Web-Sites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Web-Sites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

Lab Director
IBM Canada Ltd.
1150 Eglinton Avenue East
Toronto, Ontario M3C 1H7
Canada

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Dokument aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programme oder einer äquivalenten Vereinbarung.

Informationen über Produkte anderer Hersteller als IBM wurden von den Herstellern dieser Produkte zur Verfügung gestellt, bzw. aus von ihnen veröffentlichten Ankündigungen oder anderen öffentlich zugänglichen Quellen entnommen. IBM hat diese Produkte nicht getestet und übernimmt im Hinblick auf Produkte anderer Hersteller keine Verantwortung für einwandfreie Funktion, Kompatibilität oder andere Ansprüche. Fragen hinsichtlich des Leistungsspektrums von Produkten anderer Hersteller als IBM sind an den jeweiligen Hersteller des Produkts zu richten.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des täglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden, Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHT-LIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellensprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispielprogramme wurden nicht unter allen denkbaren Bedingungen getestet. Die in diesem Handbuch aufgeführten Beispiele sollen lediglich der Veranschaulichung und zu keinem anderen Zweck dienen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

(C) (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. (C) Copyright IBM Corp. 1997, 2001. All rights reserved. Copyright IBM Deutschland 1990, 2001 Alle Rechte vorbehalten.

Informationen zu Programmierschnittstellen

Diese Informationen sollen Ihnen bei der Erstellung von Programmen unter Verwendung dieses Produkts helfen.

Allgemeine Programmierschnittstellen ermöglichen es dem Kunden, Programme zu schreiben, die die Services der Tools dieses Programms verwenden.

Diese Informationen dokumentieren möglicherweise auch Informationen zur Diagnose, Modifizierung und Anpassung. Solche Informationen sollen Ihnen beim Debugging Ihrer Programme helfen.

Achtung: Verwenden Sie die Informationen zur Diagnose, Modifizierung und Anpassung nicht als Programmierschnittstelle, da sie ständig Änderungen unterliegen.

Marken und Dienstleistungsmarken

Die folgenden Namen sind Marken der International Business Machines Corporation in den USA und/oder anderen Ländern:

- AIX
- AS/400
- DB2
- CICS
- CICS/ESA
- IBM
- IMS
- Language Environment
- MQSeries
- Network Station
- OS/2
- OS/390
- OS/400
- RS/6000
- S/390
- VisualAge
- VTAM
- WebSphere

Lotus, Lotus Notes und Domino sind Marken oder eingetragene Marken von Lotus Development Corporation in den USA und/oder anderen Ländern.

Tivoli Enterprise Console und Tivoli Module Designer sind Marken oder eingetragene Marken von Tivoli Systems Inc. in den USA und/oder anderen Ländern.

Encina und DCE Encina Lightweight Client sind Marken oder eingetragene Marken von Transarc Corporation in den USA und/oder anderen Ländern.

Java und alle Java-basierten Marken und Logos sind Marken oder eingetragene Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.

ActiveX, Microsoft, SourceSafe, Visual C++, Visual SourceSafe, Windows, Windows NT, Win32, Win32s und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke in den USA und/oder anderen Ländern, die exklusiv über X/Open Company Limited lizenziert wird.

Intel und Pentium sind Marken von Intel Corporation in den USA und/oder anderen Ländern.

Andere Unternehmens-, Produkt- und Dienstleistungsnamen können durch einen doppelten Stern (**) gekennzeichnet sein und können Marken oder Dienstleistungsmarken anderer Unternehmen sein.